AD-A064 545 CINCINNATI UNIV OH DEPT OF ENGINEERING SCIENCE F/G 12/1
A CRITICAL EVALUATION OF COMPUTER SUBROUTINES FOR SOLVING STIFF--ETC(U)
OCT 78 D C KRINKE, R L HUSTON N00014-76-C-0139

UNCLASSIFIED UC-ES-101578-8-ONR NL

1 OF 2
AD
A064545

AD A064545

LEVEL II

DDC FILE COPY

A CRITICAL EVALUATION OF
COMPUTER SUBROUTINES FOR
SOLVING STIFF DIFFERENTIAL EQUATIONS

Dennis C. Krinke
and
Ronald L. Huston

15 Oct 78

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

DDC
RECEIVED
FEB 13 1979
E

Department of Engineering Science
Location No. 112
University of Cincinnati
Cincinnati, Ohio 45221

Technical Report under Office of Naval
Research Contract N00014-76-C-0139

79 02 09 088

See 1473

TABLE OF CONTENTS

79 02 09 088

LIST OF TABLES

ACCESSION for
NTIS White Section X
DDC Buff Section
UNANNOUNCED
JUSTIFICATION
BY
DISTRIBUTION/AVAILABILITY CODES
Dist.
A

LIST OF FIGURES

LIST OF GRAPHS

I. INTRODUCTION

The integration of initial valued simultaneous differential equations commonly occurring in models of large, complex mechanical systems (for example, human body/crash victim models, finite-segment structural system models, and large vibrating system models) has been shown to be costly in both CPU time and "turn-around" time. Indeed, developers of such models have found the efficiency of the differential equation integrators to be the most critical aspect of a model's overall efficiency [1,2,3]. Hence the search for efficient integrating subroutines or "solvers" has been long standing and is continuing.

Shampine, Watts, and Davenport [4] have made an extensive evaluation of computer codes for "nonstiff" differential equations. They suggest, however, that there are special problems which arise in the solution of "stiff" differential equations.[1] It has been observed by Huston and Hessel [5] that the differential equations of large mechanical system models are frequently stiff. Hence, there is both academic as well as utilitarian interest in a critical, comparative evaluation of the commonly used and commonly available solvers as applied to stiff equations.

Solution time and accuracy are the primary concerns when solving the differential equations of large mechanical system models. Hence,

[1] The following section of the report contains a definition of "stiffness" as associated with differential equations.

it was decided to test the available solvers on equation's where exact analytical solutions were available. However, the derivative functions (that is, "the right-hand side") of the equation for system models are generally significantly more complex than the corresponding functions for test equations for which exact solutions are known. Consequently, the subroutine which evaluates the derivative functions for system models consumes considerably more CPU time. Therefore, a measure of the efficiency of a differential equation solver is the number of function calls it needs to make to the derivative evaluating subroutine to integrate the equations while holding a given accuracy. Indeed, this is probably a better measure of the potential efficiency of a solver than the actual CPU time used to solve test equations. Thus, in the test cases considered herein, both CPU time and number of function calls are used for comparison of the solvers.[1]

The balance of this report is divided into six sections with the next section providing definitions of "stiffness". This is followed in the next three sections by a description of the subroutines tested, the test systems, and the test procedures. The final two sections contain the results and conclusions of the tests. Recommendations for further study are presented in the final section. Finally, an outline of the analytical solution of one of the test systems, together with a listing of the subroutines of the solvers tested are given in the Appendices.

---

[1]On one occasion, a program was run twice without modification and even though all other results were exactly the same, the CPU times differed by 6%. Apparently the time measurement is not as reproducible as other computer operations. Therefore when comparing these values, it is suggested that one should not expect them to be more than 5% accurate.

II. DEFINITION OF STIFFNESS

Two definitions of "stiffness" appear in literature. One defines a "stiff" linear system of differential equations as one which has widely separated eigenvalues or time constants [6,7,8,9,10]. The other associates "stiffness" with the presence of diverging exponential terms which exist in the general solution but happen to have zero coefficients in the actual solution due to the particular choice of initial conditions [11,12,13]. It is not clear that these definitions are equivalent; hence, systems which are best described by the first definition will be said to have "Type 1 stiffness" and systems which are best described by the second will be said to have "Type 2 stiffness".

Systems with Type 1 stiffness are expensive to integrate since a transient portion of the solution, which has long since decayed, prevents an increase in stepsize even though the solution at that point may be quite smooth [10,9].

Systems with Type 2 stiffness are difficult to integrate because algorithm approximation, roundoff and truncation error introduce non-zero coefficient values to the divergent exponential terms. Although these coefficients may be small, the exponential term can still grow to be large in the interval of integration [11,13]. This can, in turn, greatly reduce the accuracy of the solution.

III. SOLVER SUBROUTINE DESCRIPTIONS

Table 1. summarizes the subroutines tested and the numerical methods of each. The only subroutines tested were those that were believed to be generally available.

All of the subroutines are written in FORTRAN and are compatible with the AMDAHL 470 in use at the University of Cincinnati. DRKGS and DHPCG are interval oriented, while DVOGER, DREBS, and RK45 are step oriented. RK45 uses a constant stepsize, but all of the others use automatic stepsize adjustment. All of the routines were coded in double precision.[1]

Subroutine Name	Source	Method
DRKGS	[14]	Fourth Order Runga-Kutta
DHPCG	[14]	Hamming Predictor-Corrector
DVOGER(ADAMS)	[15]	Adams Predictor-Corrector
DVOGER(GEAR)	[15]	Gear Predictor-Corrector
DREBS	[15]	Modification of Bulirsch-Stoer ALGOL Routine DESUB
RK45	[16]	Sixth Order Runga-Kutta

Table 1. Solver Subroutine Summary

DRKGS uses a fourth order Runga-Kutta method (as modified by Gill) [14]. Some pertinent characteristics of DRKGS are as follows:

[1] RK45 was obtained in single precision, but was modified to use double precision for the test runs.

(1) Local error estimation is accomplished by comparing the solution computed in two steps of stepsize h to the solution obtained in one step of stepsize 2h;

(2) Separate error tolerances are required for each function in the system;

(3) The maximum stepsize is also used for the initial stepsize and the minimum stepsize is 2^{-10} of this value[1];

and (4) An output subroutine which is called after every step is required and is expected to perform all output duties.

DHPCG and DRKGS have been written to be easily interchanged. (The parameter lists are all identical.) To change from one subroutine to the other requires only a change in the dimension of a work array. However, DHPCG uses a completely different algorithm, that is, a Hamming predictor-corrector method [14], with a fourth order Runga-Kutta technique is used to generate the starting values. Once again, local error estimation is accomplished by comparing the solution obtained in one step of stepsize h to that obtained in two steps of stepsize h/2 and the same limitations on minimum, initial, and maximum stepsize which exist for DRKGS, exist for DHPCG.

DVOGER [15] is a modification of the subroutine DIFSUB written by C.W. Gear [17]. This routine contains two methods:

---

[1] For one of the tests, it was necessary to modify this in order to obtain a solution.

(1) An Adam's predictor-corrector method of variable order and stepsize which is intended for use with nonstiff systems;

and (2) Gear's modification of the Adam's method which is intended for use with stiff systems [18].

The pertinent characteristics of DVOGER are:

(1) Gear's method seeks to have not only the usual derivatives, but also the gradient of these functions. However, if it is not convenient to supply coding to evaluate this gradient, an option is provided that numerically approximates the gradient when given only the standard derivatives;

(2) Only a single error tolerance is permitted which is applied to all functions;

(3) Separate specifications of minimum, initial, and maximum stepsize are permitted;

and (4) Since DVOGER computes only a single step at each call, the user has the flexibility of utilizing either absolute or relative error control.

DREBS [15] is a modification of the Bulirsch-Stoer ALGOL routine DESUB. Like DVOGER, DREBS permits only one error tolerance which is applied to all functions and both absolute and relative error specifications are possible. However, although minimum and initial stepsizes may be specified, a maximum stepsize cannot be specified.

RK45 [16] computes both a fourth and a fifth order Runga-Kutta approximation to the solution at every step and then uses Richardson's

method to achieve sixth order accuracy.[1] RK45 does not estimate the local error or adjust stepsize. Furthermore, the derivative evaluating subroutine must be named XKOTEQ (although this would be simple to modify if it were inconvenient). RK45 is step oriented; however, without error control, this is no more flexible than interval orientation would be.

---

[1]The coefficients of the Runga-Kutta formulae have been optimized for maximum numerical stability.

IV. TEST SYSTEMS

Five systems of stiff differential equations were chosen to evaluate the solvers. As much as possible, systems were chosen in which the stiffness could be varied and the trend in solver efficiency observed. Systems 1, 2, and 3 contain Type 1 stiffness and systems 4 and 5 contain Type 2.

System 1 is presented in Reference [6] and represents a very simple system of differential equations that contain Type 1 stiffness. The system is:

$$\dot{\underline{y}} = [A]\,\underline{y} \tag{1}$$

where

$$[A] = \begin{bmatrix} -a & b \\ b & -a \end{bmatrix} \tag{2}$$

and

$$\underline{y}(0) = \begin{bmatrix} 0 \\ 2 \end{bmatrix} \tag{3}$$

An analytical solution is:

$$\underline{y}(t) = \begin{bmatrix} e^{-\alpha_1 t} - e^{-\alpha_2 t} \\ e^{-\alpha_1 t} + e^{-\alpha_2 t} \end{bmatrix} \tag{4}$$

where α_1(= a-b), α_2(= a+b) are the eigenvalues of A. If α_1 and α_2 are nearly equal, then the system is nonstiff, but if α_1 and α_2 have widely separated values, the system is stiff.[1] The test was conducted with the parameters shown in Table 2 and Figure 1 illustrates the solution with $\alpha_1 = 1$ and $\alpha_2 = 5$.

α_1	α_2	a	b	Period of Integration
1	2	1.5	0.5	5
1	5	3.0	2.0	5
1	10	5.5	4.5	5
1	20	10.5	9.5	5
1	50	25.5	24.5	5
1	100	50.5	49.5	5
1	200	100.5	99.5	5
1	500	200.5	249.5	5
1	1000	500.5	449.5	5

Table 2. System 1 Test Parameters.

---

[1]Interestingly, in this simple system, one can perform the transformation:

$$\underline{\omega} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \underline{y}$$

which couples the equations. That is, in terms of $\underline{\omega}$, the system becomes: $\dot{\underline{\omega}} = \begin{bmatrix} -\alpha_1 & 0 \\ 0 & -\alpha_2 \end{bmatrix} \underline{\omega}$.

With the equations decoupled, they can be solved separately and the stiffness is removed from the system.

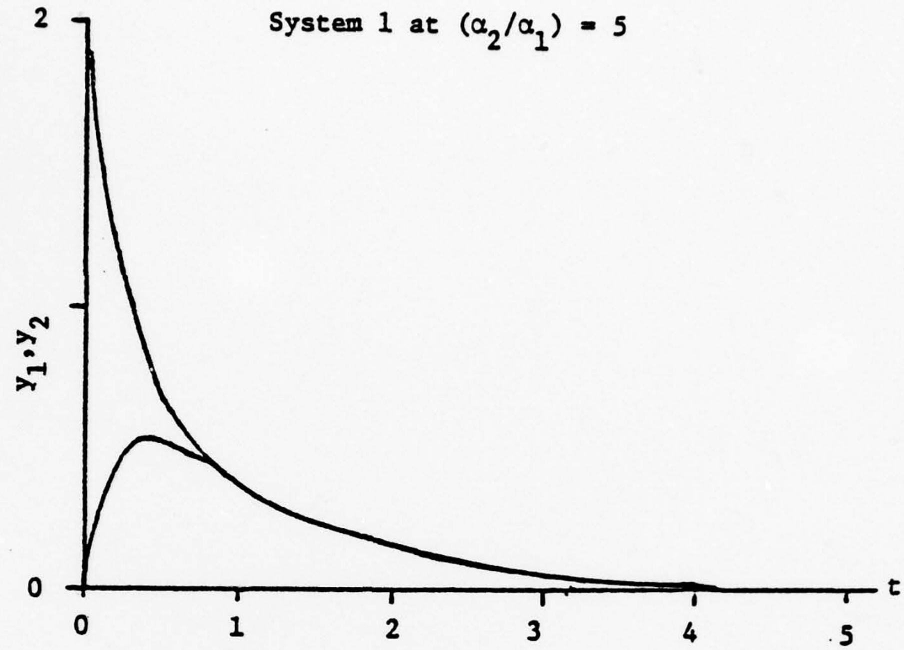


Figure 1. System 1 Exact Solution.

System 2 consists of the double mass-damper-spring system shown in Figure 2. Its governing equations are:

$$\begin{bmatrix} m_1 & 0 \\ 0 & m_2 \end{bmatrix} \ddot{\underline{y}} + \begin{bmatrix} c_1+c_2 & -c_2 \\ -c_2 & c_2 \end{bmatrix} \dot{\underline{y}} + \begin{bmatrix} k_1+k_2 & -k_2 \\ -k_2 & k_2 \end{bmatrix} \underline{y} = \underline{0} \tag{5}$$

Let the initial conditions be:

$$\underline{y}(0) = \begin{bmatrix} 1 \\ 1+k_1/k_2 \end{bmatrix} \tag{6}$$

$$\dot{\underline{y}}(0) = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{7}$$

When the system is underdamped, the analytical solution[1] has the form:

$$y_1(t) = A_1 e^{-\alpha_1 t} \cos(\omega_1 t + \phi_1) + A_2 e^{-\alpha_2 t} \cos(\omega_2 t + \phi_2) \tag{8a}$$

and

$$y_2(t) = A_3 e^{-\alpha_1 t} \cos(\omega_1 t + \phi_3) + A_4 e^{-\alpha_2 t} \cos(\omega_2 t + \phi_4) \tag{8b}$$

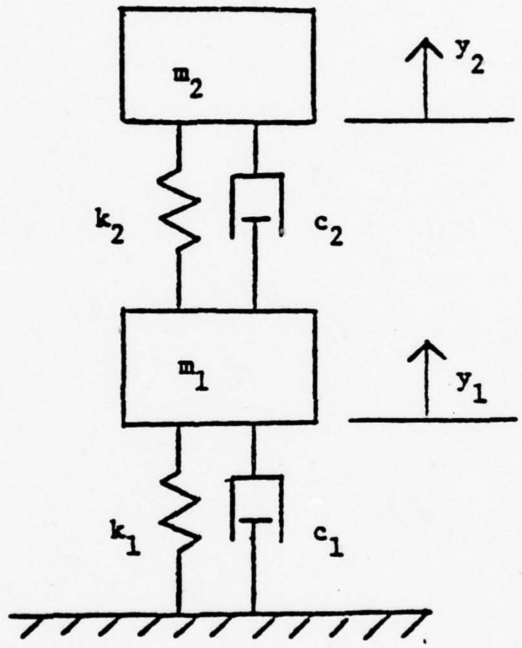


Figure 2. System 2. Model.

This solution contains damped exponential terms with time constants α_1 and α_2 . Hence, the problem is nonstiff if α_1 and α_2 are nearly equal, but has Type 1 stiffness if they have widely separated values. The system parameters m_1, m_2, c_1, c_2, k_1, and k_2 were chosen to obtain the desired values of α_1, α_2, ω_1, and ω_2. The test was conducted with

[1]The analytical solution to this system is derived in Appendix A.

the parameters shown in Table 3 and Figure 3 illustrates the solution with $\alpha_1 = 1$ and $\alpha_2 = 5$.

α_1	α_2	ω_1	ω_2	Period of Integration
1	2	2π	6π	5
1	5	2π	6π	5
1	10	2π	6π	5
1	20	2π	6π	5
1	50	2π	6π	5
1	100	2π	6π	5
1	200	2π	6π	5
1	500	2π	6π	5
1	1000	2π	6π	5

TABLE 3. System 2 Test Parameters.

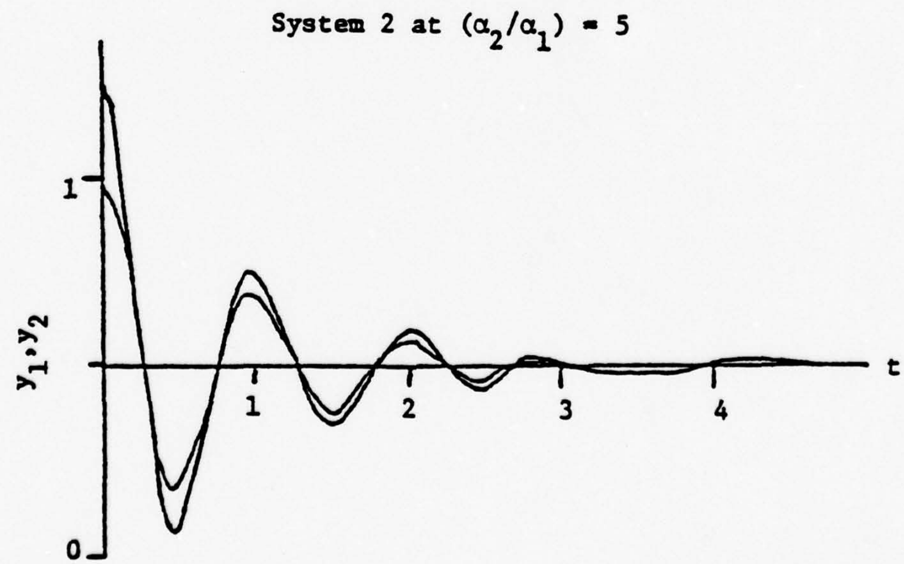


Figure 3. System 2 Exact Solution.

System 3 was obtained from References [6,17] and is attributed to F. T. Krogh. The equation

$$\dot{y} = -\beta\, y + y^2 \quad \text{with} \quad y(0) = -1 \tag{9}$$

has the analytical solution:

$$y(t) = \frac{\beta}{1-(1+\beta)e^{\beta t}} \tag{10}$$

If one takes the system of four uncoupled equations:

$$\dot{y}_i = -\beta_i\, y_i + y_i^{\,2} \;, \tag{11}$$

$$y_i(0) = -1, \quad i = 1,2,3,4 \tag{12}$$

and performs the transformation:

$$\underline{w} = [U]\, \underline{y} \tag{13}$$

where

$$[U] = \frac{1}{2}\begin{bmatrix} -1 & 1 & 1 & 1 \\ 1 & -1 & 1 & 1 \\ 1 & 1 & -1 & 1 \\ 1 & 1 & 1 & -1 \end{bmatrix} \tag{14}$$

then the system becomes

$$\dot{\underline{w}} = -[U]\ [B]\ [U]\underline{w} + [u]\ \underline{Z} \tag{15}$$

with $\quad w_i(0) = -1 \quad i = 1,2,3,4$ (16)

where $$[B] = \begin{bmatrix} \beta_1 & 0 & 0 & 0 \\ 0 & \beta_2 & 0 & 0 \\ 0 & 0 & \beta_3 & 0 \\ 0 & 0 & 0 & \beta_4 \end{bmatrix} \tag{17}$$

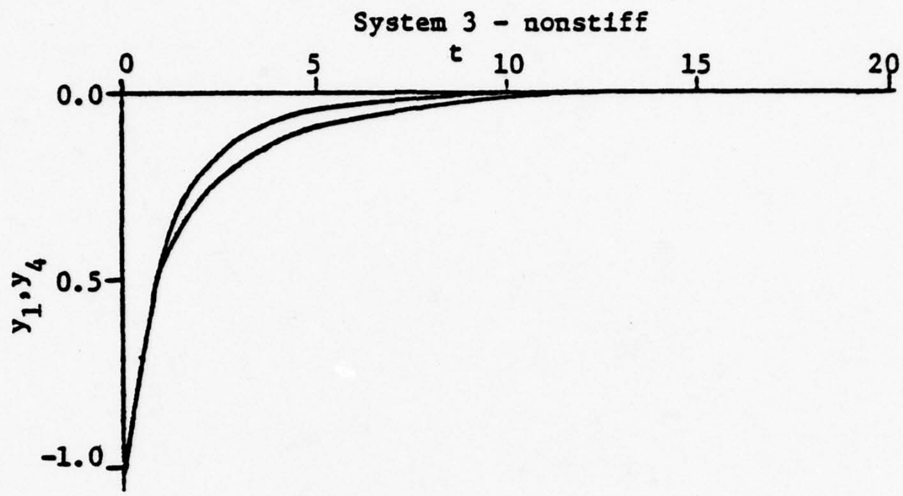
and $Z_i = w_i^2$ (18)

In this form, the equations are coupled and have Type 1 stiffness. Krogh suggests letting the values for the β_i's be 0.1, 0.2, 0.3, and 0.4 for a nonstiff problem and 1000, 800, -10, and 0.001 for a stiff problem. The test was conducted with both sets of these values and Figure 4 illustrates the nonstiff solution. The period of integration was 20 for the nonstiff problem and was 1000 for the stiff problem.

System 4 is presented in Reference [11] and consists of the single equation:

$$\dot{y} = 5(y - t^2) \quad t\varepsilon(0,5)\ . \tag{19}$$

The general analytical solution is:



$$y(t) = C\,e^{5t} + t^2 + 0.4t + 0.08 \quad . \tag{20}$$

In the special cases where $y(0) = 0.08$, the value of C is exactly zero. However, algorithm approximation, roundoff and truncation errors cause the numerical solution to contain a ke^{5t} term. k must be very small indeed for ke^{5t} to be small in comparison to the exact solution, which is shown in Figure 5.

System 5 was a clear example of Type 2 stiffness but it lacked a means of varying that stiffness. System 5 does not obviously have Type 2 stiffness, but it behaves in sufficiently similar manner to suggest that it does.

System 5 models a central force orbit (two body problem with the mass of one body much larger than the other). Elliptical orbits

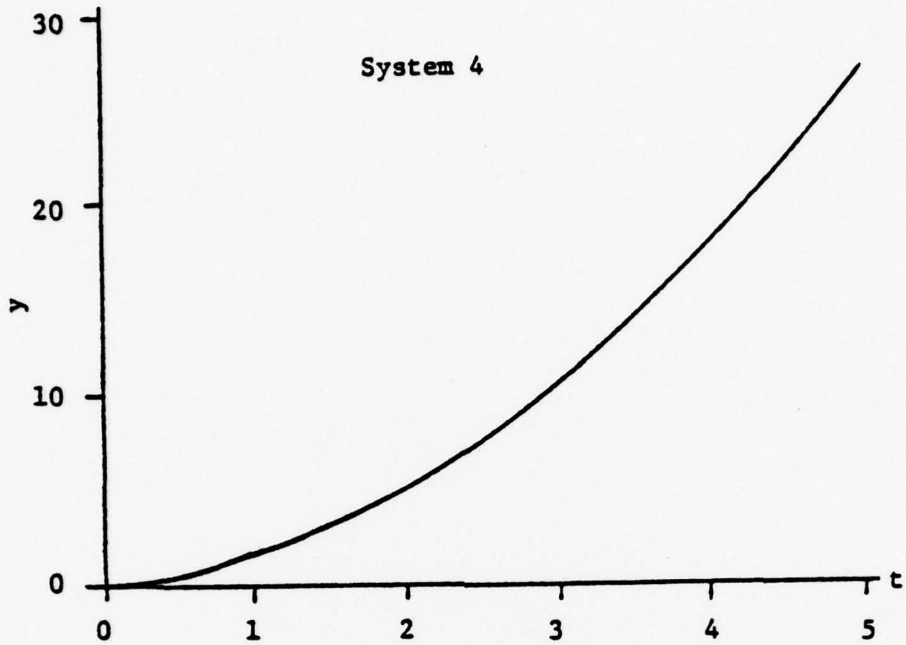


Figure 5. System 4 Exact Solution.

were chosen so that they would be periodic. The eccentricity of the ellipse was varied from moderately elliptical to highly "cigar-shaped" in order to change the amount of stiffness. The governing equations for this system are:

$$\ddot{r} = r\dot{\theta}^2 - \frac{GM}{r^2} \tag{21a}$$

$$\ddot{\theta} = -2\dot{r}\dot{\theta}/r \tag{21b}$$

where G and M are constants.

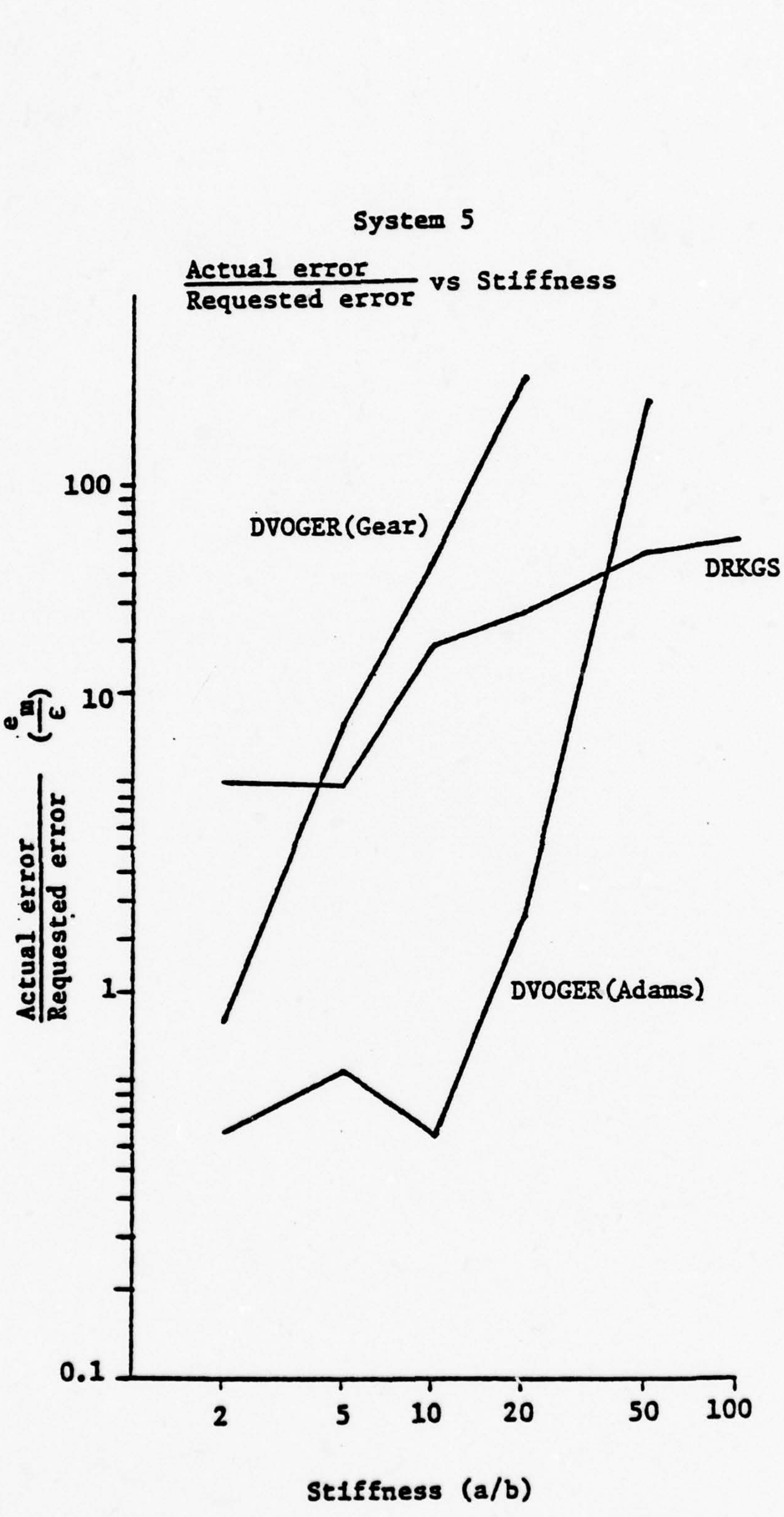
The initial conditions (which correspond to the apogee) are:

$$r(0) = a(1+e) \tag{22a}$$

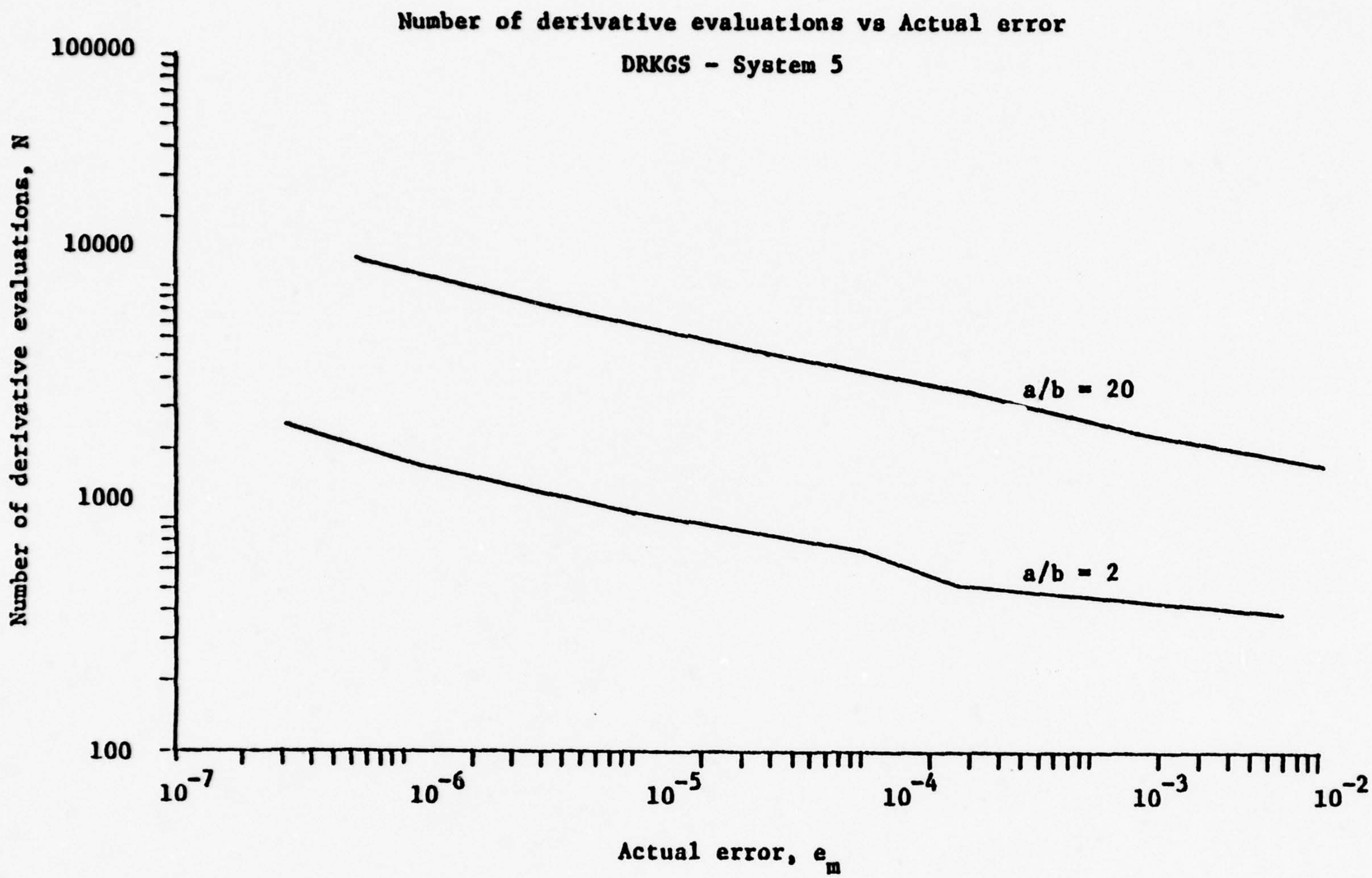
$$\theta(0) = -\pi \tag{22b}$$

System 5

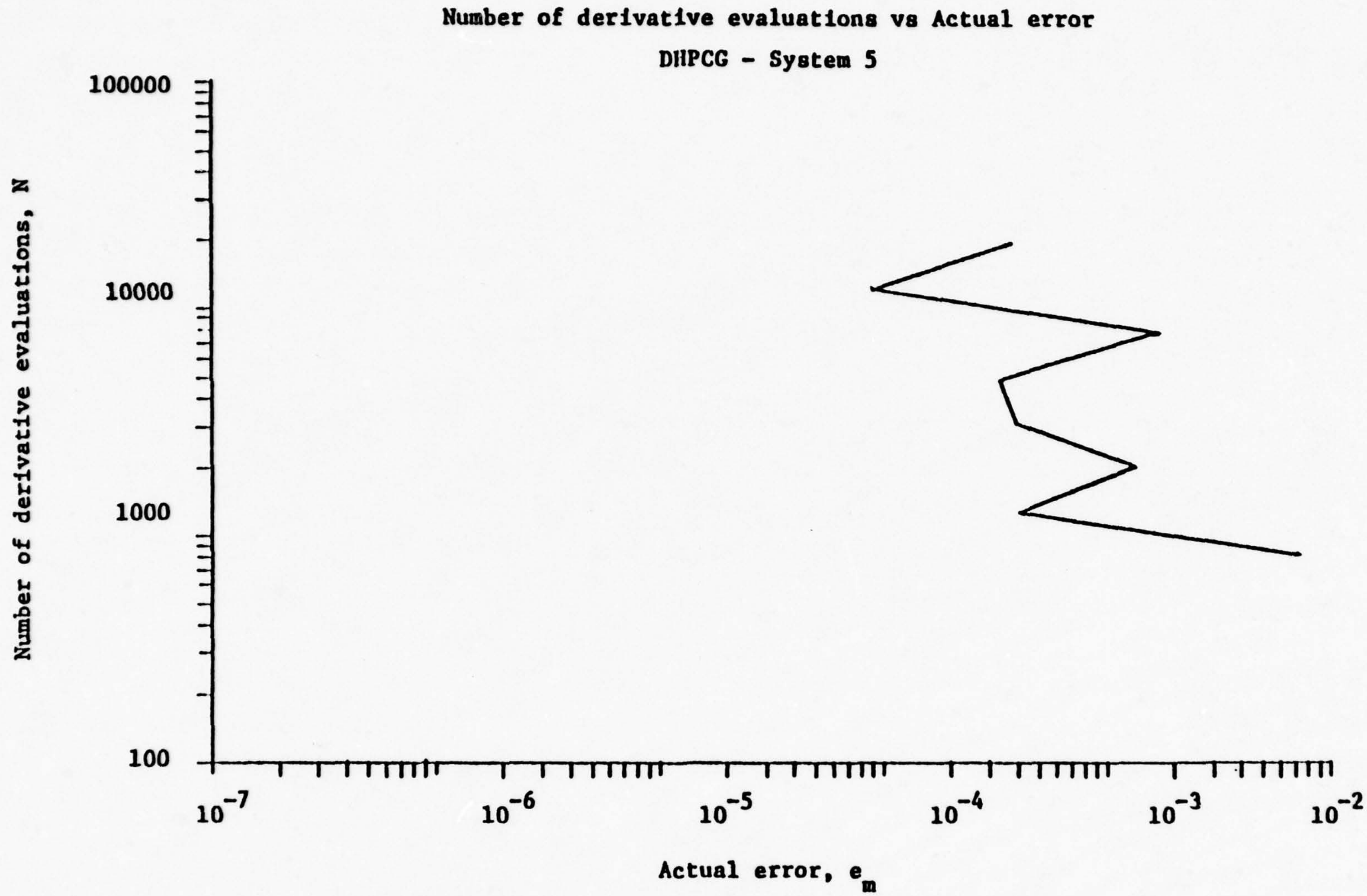
$\frac{\text{Actual error}}{\text{Requested error}}$ vs Stiffness



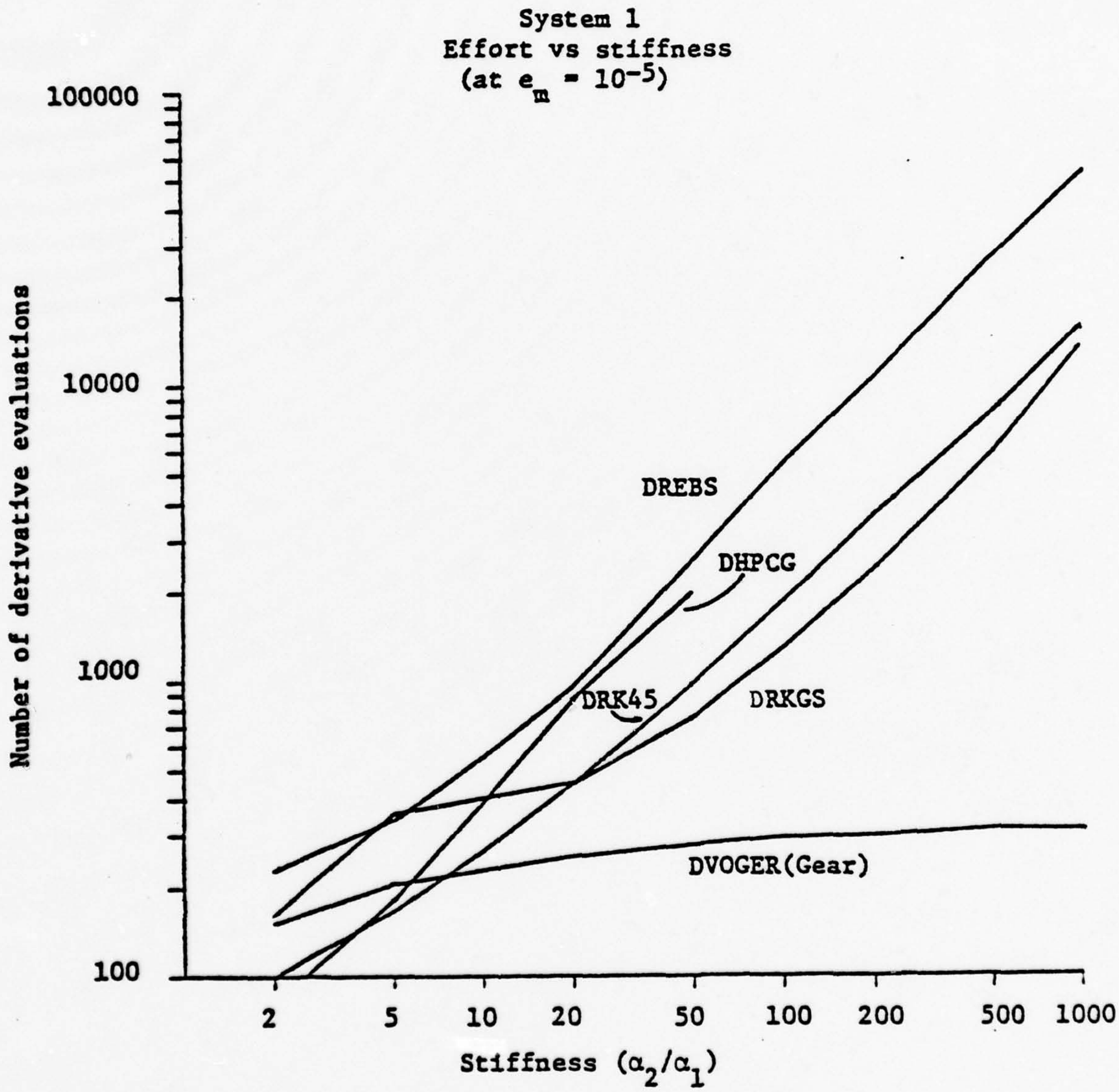
Graph 5. Accuracy vs. Stiffness - System 5.



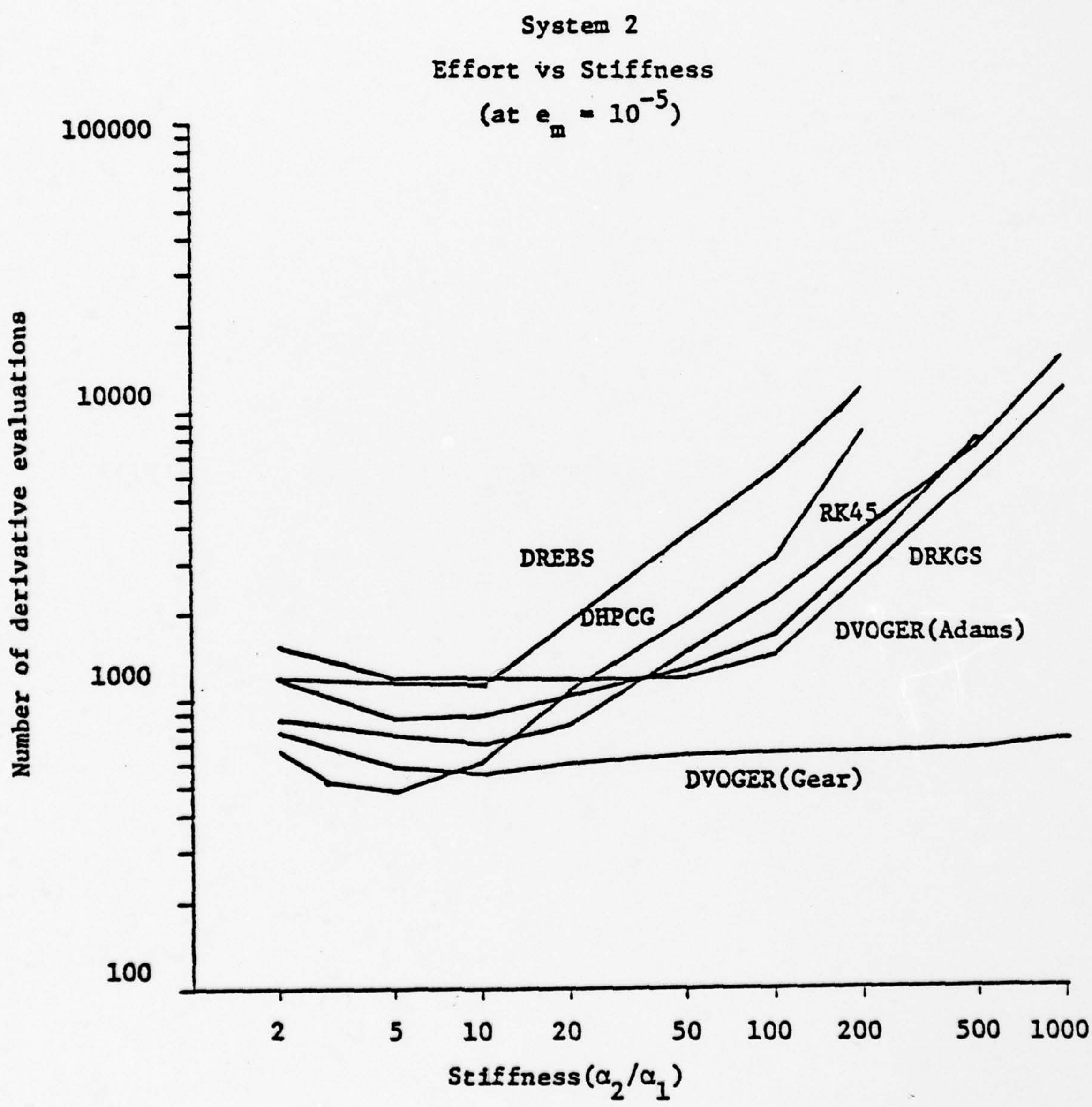
Graph 6. Effort vs. Accuracy - System 5 - DRKGS



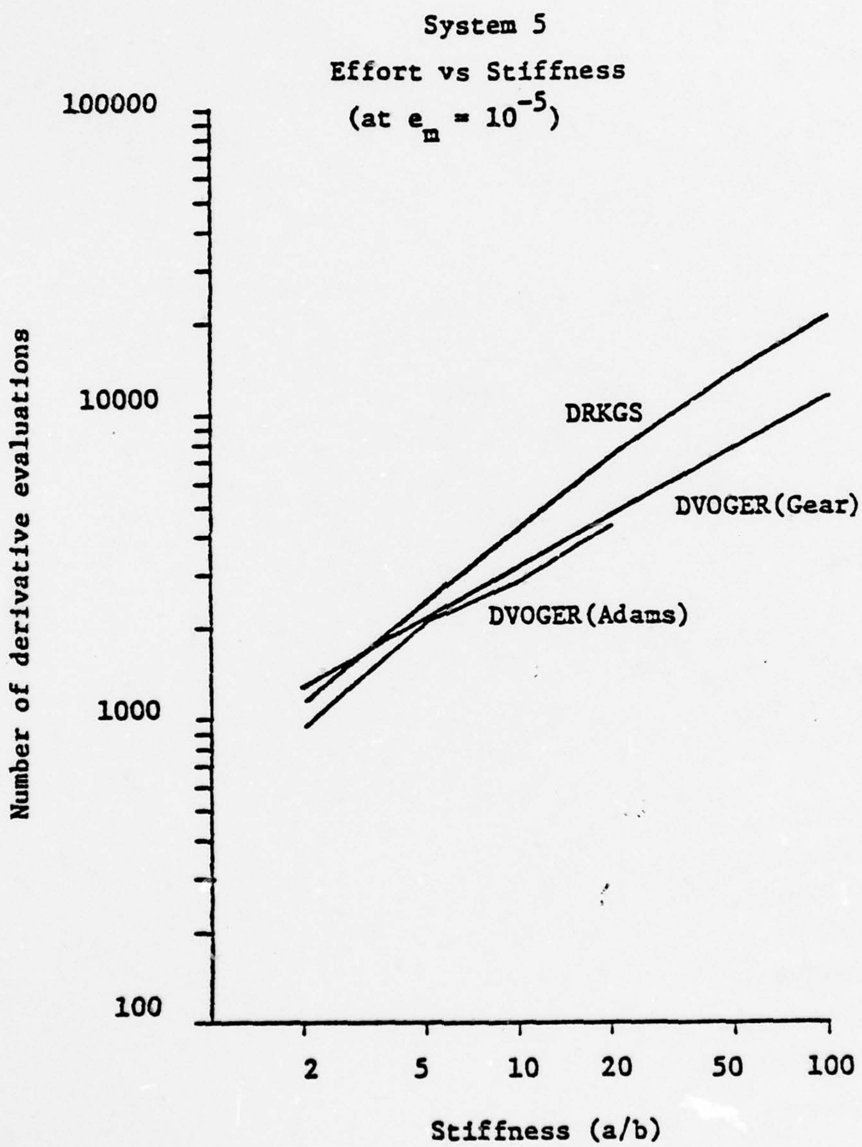
Graph 7. Effort vs. Accuracy - System 5 - EHPCG



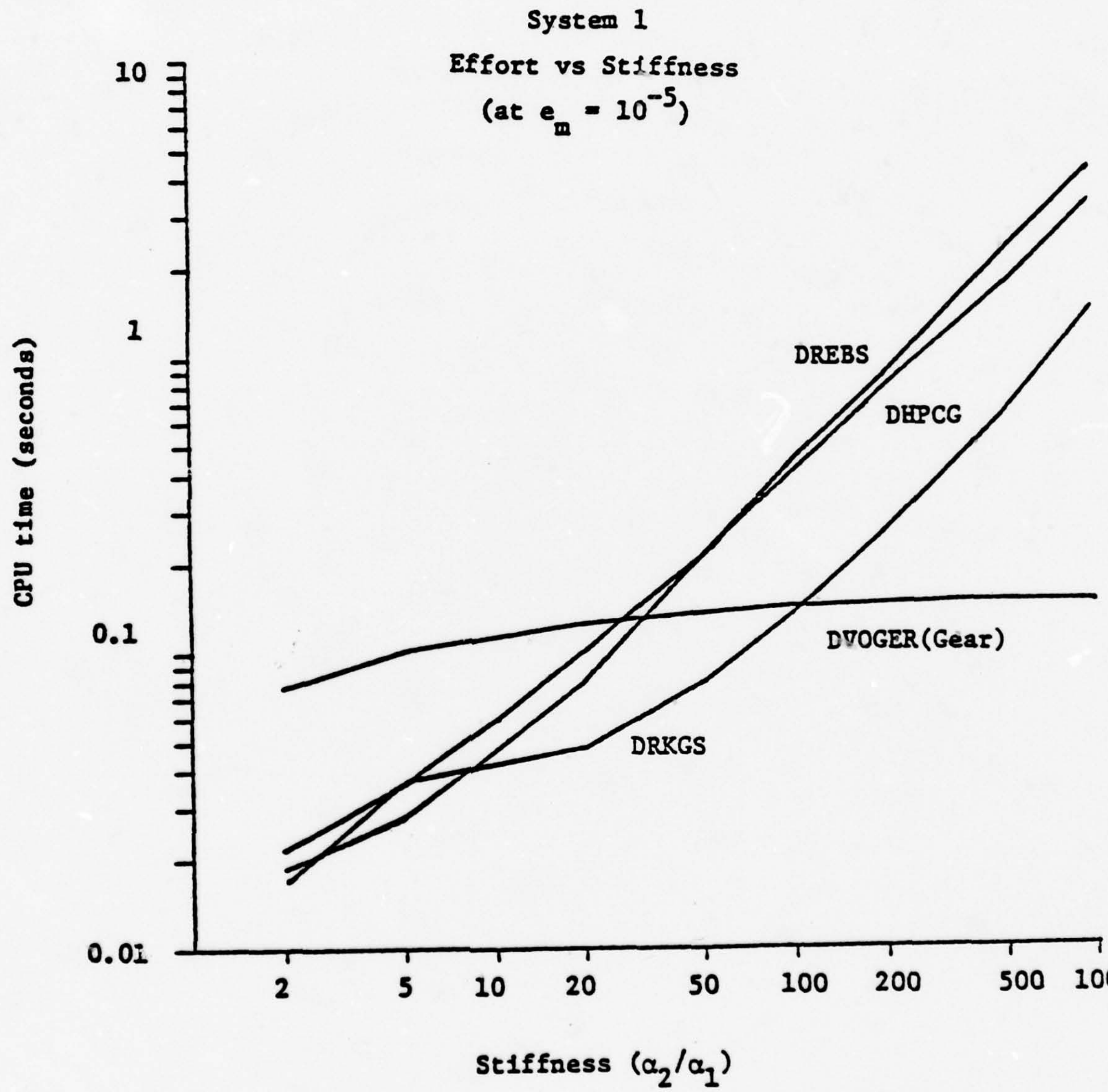
Graph 8. Effort vs. Stiffness - System 1.



Graph 9. Effort vs. Stiffness - System 2.



Graph 10. Effort vs. Stiffness - System 5.



Graph 11. CPU Time vs. Stiffness - System 1.

VII. CONCLUSIONS AND RECOMMENDATIONS

Two significant and important conclusions can be drawn from this research effort. First, the solvers may not yield the requested accuracy when integrating a system with Type 2 stiffness. This suggests that the integration of such systems is not automatic at all. Rather, one must integrate every such system at least twice with different requested error tolerances (ε) and compare solutions. While such stiff systems are not as common as non-stiff systems, the user should be cautioned that they do exist.

Secondly, if a system has Type 1 stiffness and if the derivative evaluations are expensive, a routine using Gear's method (such as DVOGER) should be used. Gear's method was designed especially for stiff systems and it can be strikingly more efficient than the other solver routines.

Beyond this, experience with these subroutines prompts a few other remarks and opinions: First, even in system that were not stiff, the actual maximum error (e_m) frequently differed from that requested (ε) by factors of 20 or more. The subroutines would be more satisfying if their global error estimation could be improved.

Secondly, RK45 does not have automatic stepsize capability and was difficult to use. This emphasized the convenience of the automatic stepsize adjustment of the other subroutines. Even if one must solve every problem twice with them it is much better than making four or five runs to find the correct stepsize for RK45.[1]

[1] When the stepsize is too large for RK45, numerical instability occurs, leading to exponential overflow.

Third, the interval oriented format of DRKGS and DHPCG was convenient. Also, placing all of the output duties into a subroutine gives a modular property to the coding and further simplifies their use. However, the inability to separately specify the initial and maximum value of stepsize in these routines is disasterous to efficiency when looking for long time solutions in systems with decaying transients. Also, it is sometimes necessary to extend the permitted number of stepsize bisections to obtain solutions. Ten bisections may be a reasonable limit for single precision work, but forty or even fifty can be required for double precision integrations.

Finally, DVOGER and DREBS have no provision for the specification of separate error tolerances for the different functions in the system. While this presented no serious difficulty for this work, it could conceivably be important for a system with functions of greatly different magnitudes. The step oriented format of DVOGER and DREBS does however possess the potential for greater flexibility, particularly in error control. For example, if one suspected Type 2 stiffness, an exponential distribution of error per step would probably be better than a linear distribution.

REFERENCES

1. Karnes, R.N., Tocher, J.L. and Twigg, D.W., "PROMETHEUS-A Crash Victim Simulator," Aircraft Crashworthiness, University Press of Virginia, 1975, pp. 327-345.

2. Fleck, J.T., Butler, F.E. and Vogel, S.L., "An Improved Three-Dimensional Computer Simulation of Motor Vehicle Crash Victims," Report CAL No. ZQ-5180-L-1, Calspan Corp., Buffalo, N.Y., 1974.

3. Huston, R.L., Hessel, R.E. and Winget, J.M., "Dynamics of a Crash Victim—A Finite Segment Model," AIAA Journal, Vol. 14, No. 2, Feb. 1976, pp. 173-178.

4. Shampine, L.F., Watts, H.A. and Davenport, S.M., "Solving Nonstiff Ordinary Differential Equations-The State of the Art," SIAM Review, Vol. 18, 1976, pp. 376-411.

5. Huston, R.L. and Hessel, R.E., private communication.

6. Lapidus, L. and Schiesser, W.E., Numerical Methods for Differential Systems, Academic Press Inc., New York, 1976.

7. Willoughby, R.A., Stiff Differential Systems, Plenum Press, New York, 1974.

8. Lapidus, L. and Seinfeld, J.H., Numerical Solution of Ordinary Differential Equations, Academic Press, New York, 1971.

9. Shampine, L.F. and Gordon, M.K., Computer Solution of Ordinary Differential Equations: The Initial Value Problem, W.H. Freeman, San Francisco, 1975.

10. Forsythe, G.E., Malcolm, M.A. and Moler, C.B., Computer Methods for Mathematical Computations, Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1977.

11. Hornbeck, R.W., Numerical Methods, Quantum Publishers Inc., 1975.

12. Gerald, C.F., Applied Numerical Analysis, Addison-Wesley Publishing Co., Reading, Massachusetts, 1978.

13. Acton, F.S., Numerical Methods that Work, Harper and Row, New York, 1970.

14. IBM Scientific Subroutine Package-Programmer's Manual, IBM, White Plains, New York, 1966.

15. International Mathematical and Statistical Libraries Reference Manual, Houston, Texas, 1977.

16. Villadsen, J. and Michelson, M.L., Solution of Differential Equations by Polynomial Approximation, Prentice-Hall, Englewood Cliffs, New Jersey, 1978.

17. Gear, C.W., "Algorithm 407 DIFFSUB for Solution of Ordinary Differential Equations", Comm. ACM, Vol. 14, 1971, pp. 185-187.

18. Gear, C.W., "The Automatic Ingetration of Ordinary Differential Equations," Comm. ACM, Vol. 14, 1971, pp. 176-179.

19. Kane T.R., Dynamics, Stanford University, 1972, pp. 205-207.

$$\dot{r}(0) = 0 \tag{22c}$$

$$\dot{\theta}(0) = 2\pi ab/(r^2(0)T) \tag{22d}$$

where a and b are respectively the major and minor semi-axes of the ellipse, e is the eccentricity ($\sqrt{1-(b/a)^2}$), and T is the period of the orbit. To achieve a period T, it is necessary to set $GM = a^3(2\pi/T)^2$. The exact solution is simple only at $t = T$:

$$r(T) = r(0) \tag{23a}$$

$$\theta(T) = \pi \tag{24b}$$

The test was conducted with the parameters given in Table 4 and Figure 6 illustrates the solution for $a/b = 2$.

a	a/b	T
5	2	1
5	5	1
5	10	1
5	20	1
5	50	1
5	100	1

TABLE 4. System 5 Test Parameters.

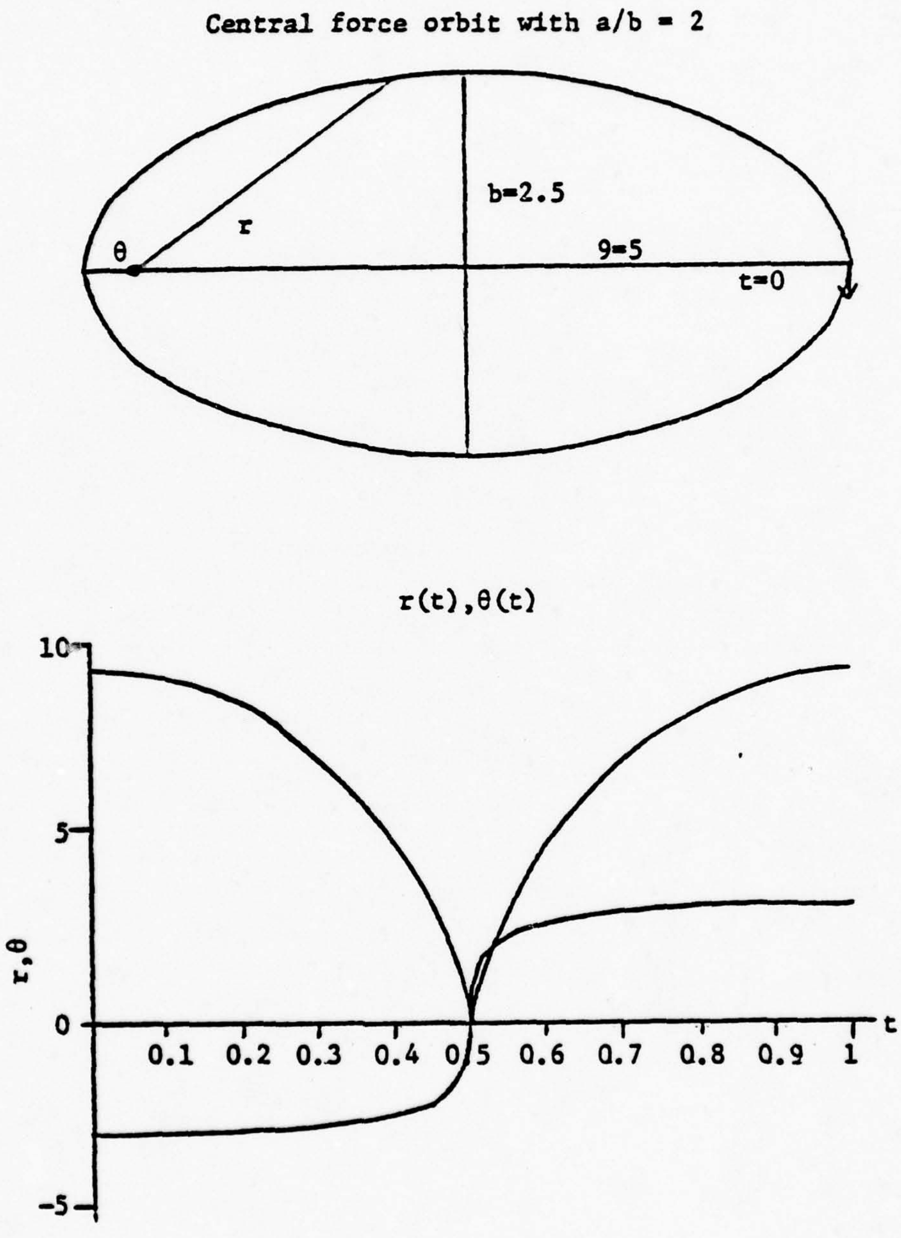


Figure 6. System 5 Exact Solution.

V. TEST PROCEDURE

In all of the tests, the number of derivative evaluating subroutine calls (N), the CPU time of integration, and the maximum occurring error ($error_{max}$) were measured. (The solver DVOGER was used with both Adam's and Gear's methods and when it was used with Gear's method, the option was utilized in which the gradient of the derivative functions was evaluated numerically.)

All of the solvers, except RK45, require that an error tolerance (EPS) be given. For systems 1, 2, and 3, each integration was performed with EPS = 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}. For system 4, this was extended to 10^{-12} and for system 5 to 10^{-8}. RK45 was executed with a range of stepsizes which produced similar accuracy.

For the solvers which require a minimum stepsize, 10^{-15} of the interval value was used. This corresponds to about 140 units of round-off.

All tests were executed in double precision.

VI. RESULTS AND DISCUSSION

As mentioned previously, the primary interests are solver efficiency and accuracy. To discuss efficiency, it is desirable to consider the efficiency at a specified accuracy. Hence accuracy is discussed first.

The documentation of DRKGS and DHPCG state that the maximum global error, e_m, is usually of about the same magnitude as the specified error tolerance, ε , but the documentations of the other solvers do not state that. To access the accuracy of the solvers e_m vs. ε was plotted for several cases. Graphs 1 and 2 are typical of these results and deal with systems 1 and 5 respectively. Error estimation is clearly not exact and it is somewhat disappointing that e_m differed by a factor of 20 or 50 from ε so frequently. From graph 1 alone, one might think that DHPCG is much superior to the other solvers in error estimation, but this is merely coincidence. There are other graphs where it appears to be poor and another appears to be good.

These graphs show only one value of stiffness and do not show if stiffness influences solver accuracy. Systems 1, 2 and 5 were specifically chosen because their stiffness could be varied in a desired manner. Graphs 3, 4 and 5 show explicitly how stiffness influences solver accuracy. These are graphs of (e_m/ε) vs. stiffness. Graphs 3 and 4 deal with systems 1 and 2 respectively and hence Type 1 stiffness. Graph 5 deals with System 5 and Type 2 stiffness. In Graph 3, it appears that DREBS and DRKGS lose accuracy with increasing stiffness,

but that behavior is not confirmed by Graph 4. The other routines are not clearly affected by Type 1 stiffness and DVOGER appears to be the least influenced. However, a look at Graph 5 clearly shows that Type 2 stiffness is different. As the stiffness of the problem increases, e_m grows much larger than ε.

Systems 3 and 4 contain Type 1 and Type 2 stiffness, respectively, and are even stiffer than the most stiff cases of systems 1, 2, and 5. Table 5 contains data pertinent to solver accuracy for systems 3 and 4. Note that e_m can be many orders of magnitude greater than ε.

(e_m/ε) for very stiff systems		
	SYSTEM 3 (Type 1)	SYSTEM 4 (Type 2)
DRKGS	-	1.2×10^{12}
DVOGER(GEAR)	4.6	3.1×10^{10}
DVOGER(ADAMS)	-	1.8×10^{10}
DREBS	-	1.3×10^{8}
DHPCG	-	2.3×10^{8}

Table 5. Errcr Ratio Data.

In every test integration, different solvers generated different accuracies, even when ε was constant and hence, it is probably misleading to compare solver effort at equal ε. Rather, it is probably more appropriate to compare solver effort at equal values of e_m. Graphs of N (the number of calls to the derivative evaluating subroutine) vs. e_m were prepared and estimates of N at a specific e_m were obtained. Usually these graphs were smooth and appeared to be hyperbolic (straight lines on log-log plots). Typical of these results is Graph 6 which pertains to System 5 and DRKGS. However, in some cases, these graphs were not smooth and Graph 7, concerning System 5 and DHPCG, shows such a result. In these cases, no attempt was made to estimate N.

In most of the tests, the solvers achieved 10^{-5} accuracy and this value of e_m was chosen to compare solver efficiencies. Graphs 8, 9, and 10 pertain to Systems 1, 2, and 5, respectively. They show how stiffness influences integration effort. The effect is quite striking. The systems in Graphs 8 and 9 both contain Type 1 stiffness. Almost every routine requires more effort for the integration as stiffness increases. However, DVOGER (Gear) was specifically designed for systems with this type of stiffness and it is relatively unaffected by Type 1 stiffness. System 5 in Graph 10 contains Type 2 stiffness and it is clear that this stiffness increases integration effort. No routine was much more efficient than another in this problem. Rather, the true test was whether they could integrate all of the cases. Only DRKGS and DVOGER (Gear) achieved solutions accurate to 10^{-5} for all stiffnesses.

System 3, which had very much Type 1 stiffness could be solved to 10^{-5} accuracy only by DVOGER (Gear). The other routines took so many

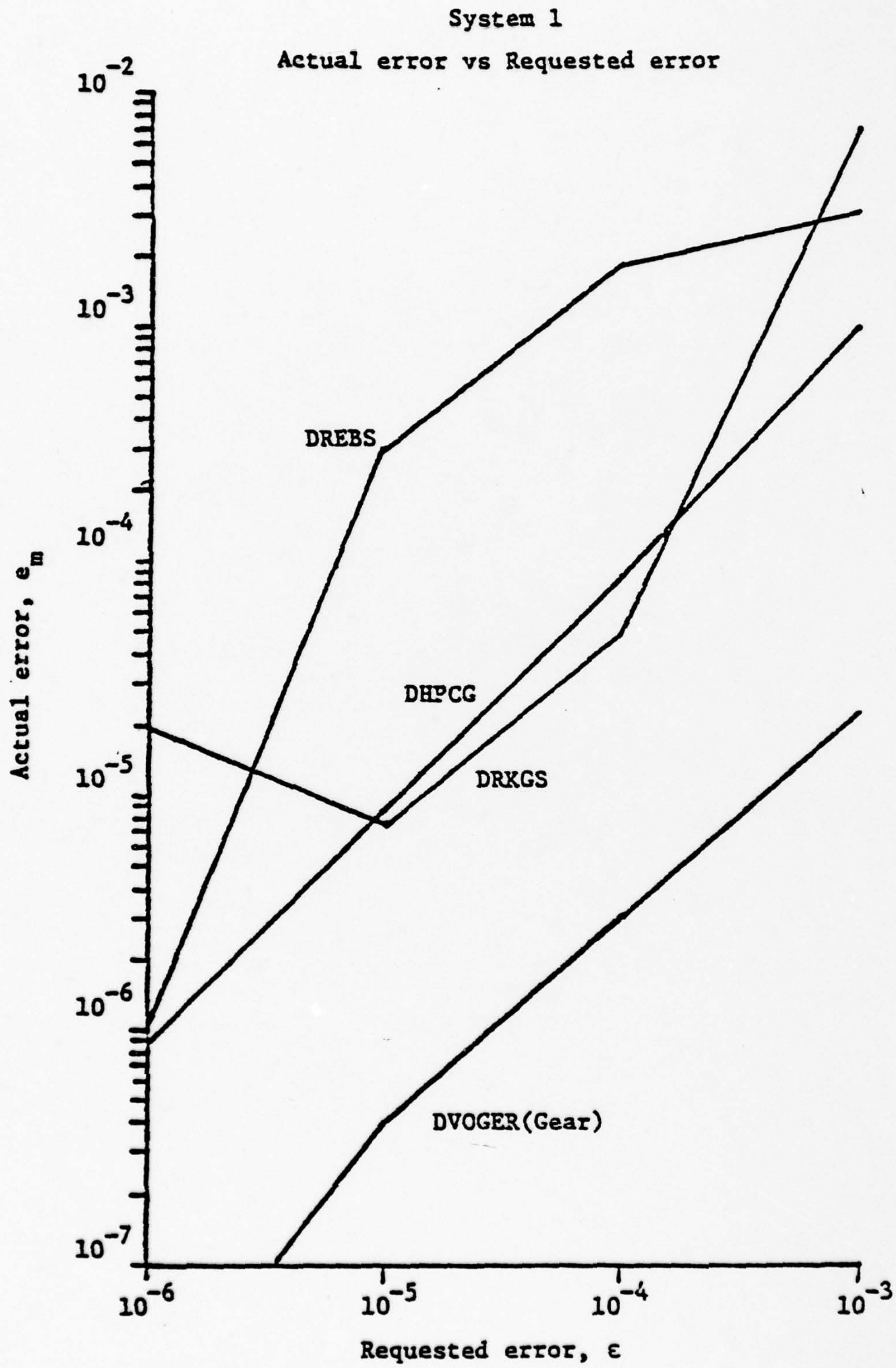
function evaluations that apparently roundoff error limited their accuracy. System 4, which had very much type 2 stiffness, was solved by DRKGS, DRK45, DREBS and DHPCG to 10^{-3} relative error or better (10^{-2} was considered to be the *minimum* acceptable relative error). While DVOGER did not achieve this accuracy with absolute error control, it might perform better with relative error control. DHPCG achieved the best accuracy, about 10^{-6}.

The CPU times of integration, T, were measured but they primarily represent overhead (time spent in the solver subroutine rather than the derivative evaluating subroutine). The overhead per derivative evaluation was computed and these values are shown in Table 6. For systems where the derivative evaluations are simple, these values of (T/N) may be used to convert results from N to T

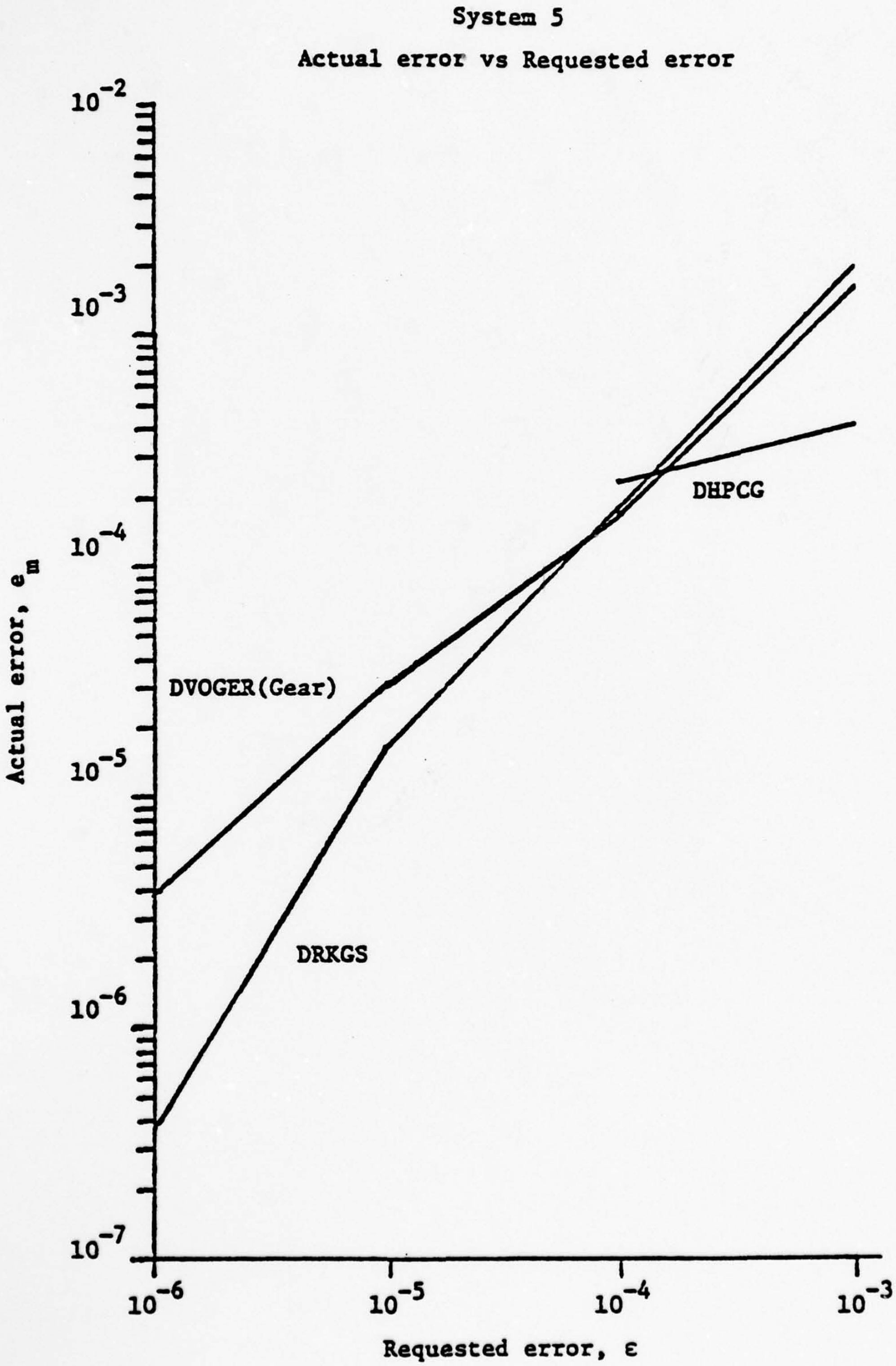
Graph 11 for the data from Graph 8 (System 1).

ROUTINE	CPU TIME PER DERIVATIVE SUBROUTINE CALL (μ-SEC)
DREBS	80
DRK45	103
DRKGS	106
DHPCG	220
DVOGER(ADAMS)	400
DVOGER(GEAR)	500

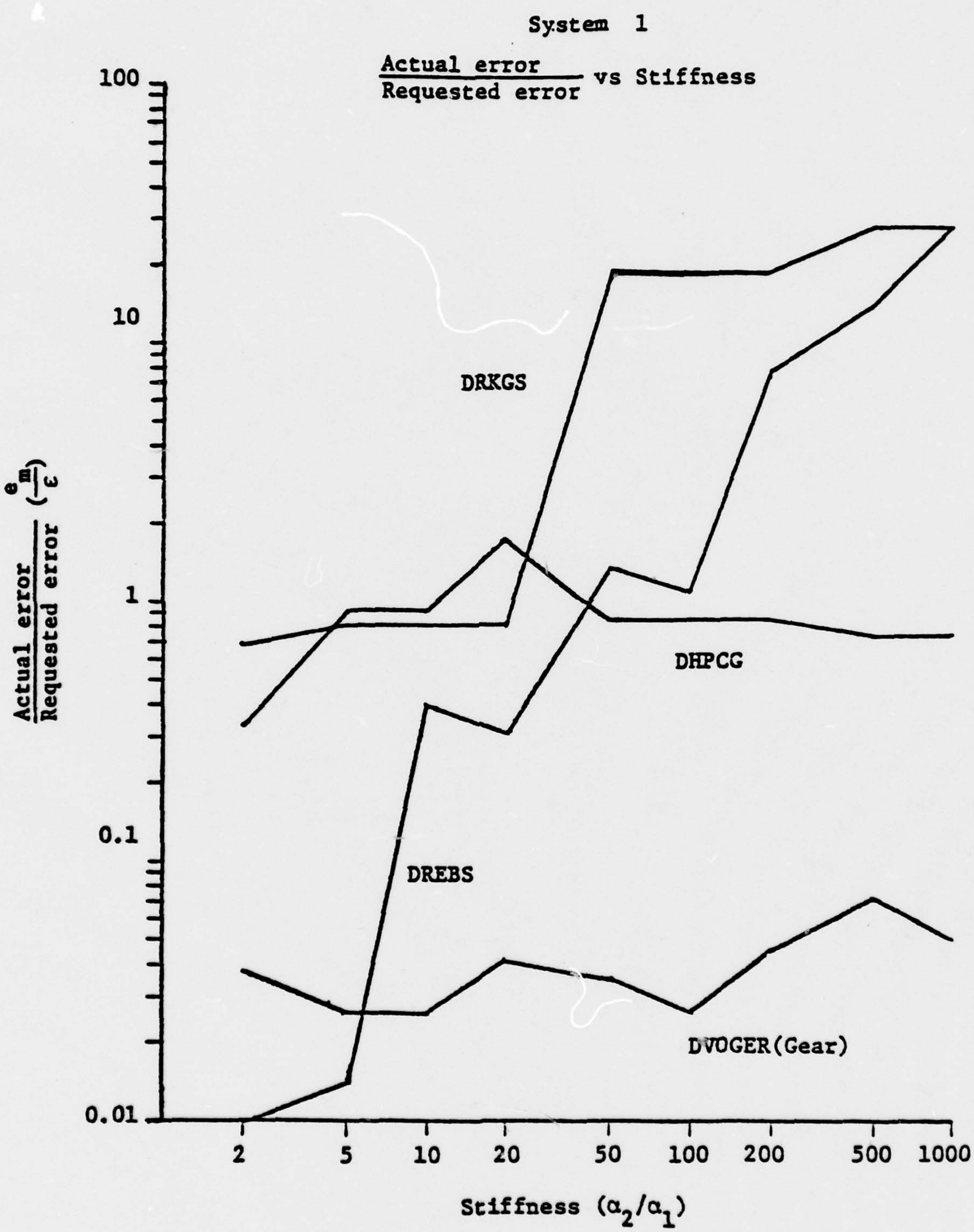
TABLE 6. CPU for Derivative Evaluation.



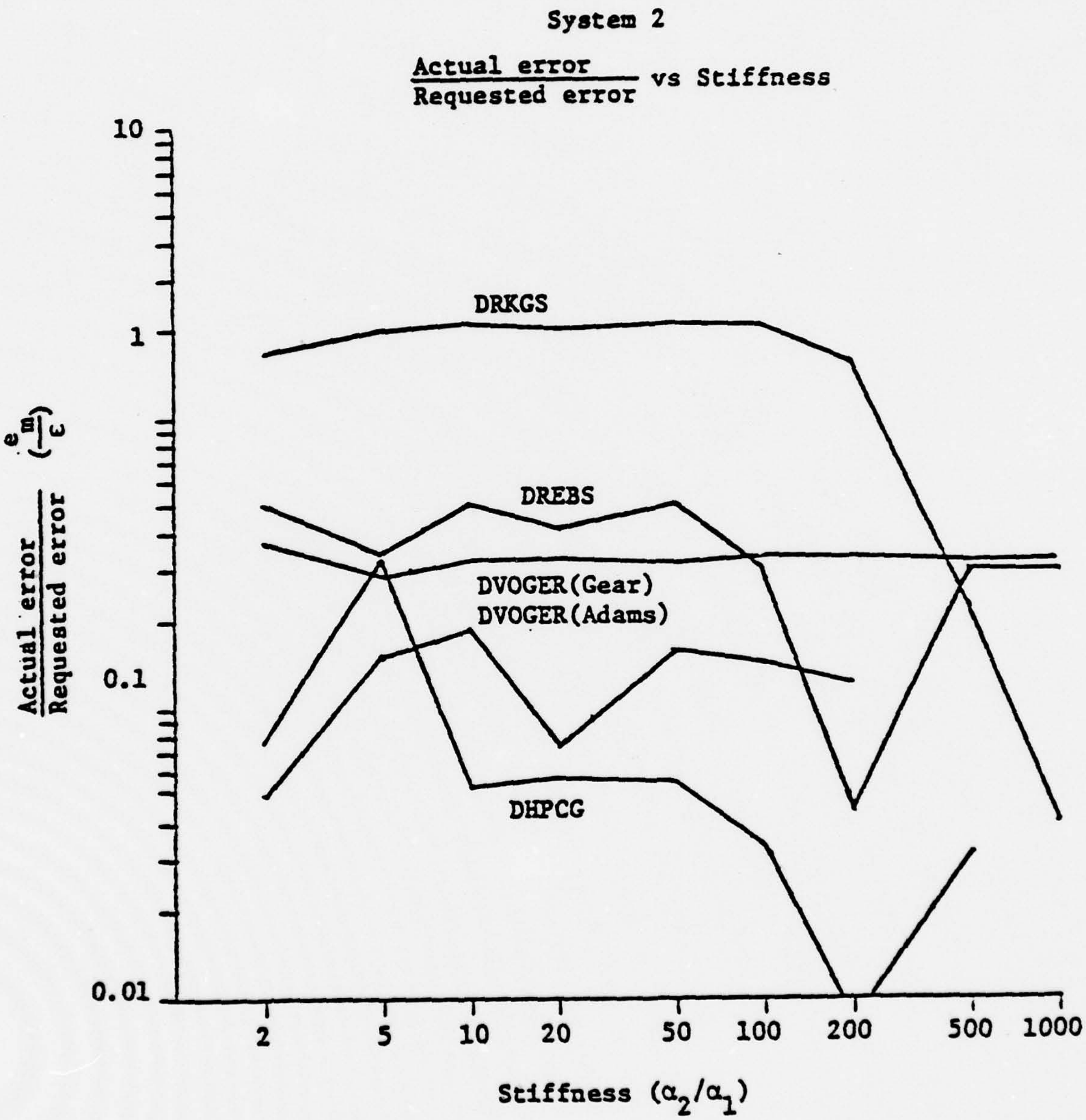
Graph 1. Actual Error vs. Requested Error - System 1.



Graph 2. Actual Error vs. Requested Error - System 5.



Graph 3. Accuracy vs. Stiffness - System 1.



Graph 4. Accuracy vs. Stiffness - System 2.

APPENDIX A

EXACT SOLUTION TO THE DOUBLE-MASS-SPRING DASHPOT SYSTEM (SYSTEM 2)

The governing differential equations of motion for the system are:

$$m_1\ddot{y}_1 + (c_1 + c_2)\dot{y}_1 - c_2\dot{y}_2 + (k_1 + k_2)y_1 - k_2y_2 = 0 \qquad \text{(A1)}$$

and

$$m_2\ddot{y}_2 \quad - c_2\dot{y}_1 + c_2\dot{y}_2 \quad -k_1y_1 + k_2y_2 = 0 \qquad \text{(A2)}$$

Taking the Laplace transform of these equations and solving for $y_1(s)$ and $y_2(s)$ leads to:

$$y_1(s) = \frac{a_3s^3 + a_2s^2 + a_1s + a_0}{d_4(s^4 + d_3s^3 + d_2s^2 + d_1s + d_0)} \qquad \text{(A3)}$$

$$y_2(s) = \frac{b_3s^3 + b_2s^2 + b_1s + b_0}{d_4(s^4 + d_3s^3 + d_2s^2 + d_1s + d_0)} \qquad \text{(A4)}$$

where:

$$a_0 = c_1k_2\ y_1(0) + m_1k_2\ \dot{y}_1(0) + m_2k_2\ \dot{y}_2(0) \tag{A5}$$

$$a_1 = (m_1k_2 + c_1c_2)y_1(0) + m_2k_2\ y_2(0) + m_1c_2\ \dot{y}_1(0) + m_2c_2\dot{y}_2(0) \tag{A6}$$

$$a_2 = [m_1c_2 + m_2(c_1 + c_2)]y_1(0) + m_1m_2\ \dot{y}_1(0) \tag{A7}$$

$$a_3 = m_1m_2\ y_1(0) \tag{A8}$$

$$b_0 = (c_1k_2 - c_2k_1)y_1(0) + c_2k_2\ y_2(0) + m_1k_2\ \dot{y}_1(0) + m_2(k_1 + k_2)\dot{y}_2(0) \tag{A9}$$

$$b_1 = m_1k_2\ y_1(0) + [m_2(k_1 + k_2) + c_1c_2)y_2(0) + m_1c_2\ \dot{y}_1(0) + + m_2(c_1 + c_2)\dot{y}_2(0) \tag{A10}$$

$$b_2 + -m_1c_2\ y_1(0) + [m_1c_2 + m_2(c_1 + c_2)]y_2(0) + m_1m_2\ \dot{y}_2(0) \tag{A11}$$

$$b_3 = m_1m_2\ y_2(0) \tag{A12}$$

$$d_0 = k_1k_2/d_4 \tag{A13}$$

$$d_1 = (c_1k_2 + c_2k_1)/d_4 \tag{A14}$$

$$d_2 = [m_1 k_2 + m_2(k_1 + k_2) + c_1 c_2]/d_4 \tag{A15}$$

$$d_3 = [m_1 c_2 + m_2(c_1 + c_2)]/d_4 \tag{A16}$$

$$d_4 = m_1 m_2 \tag{A17}$$

To "invert" the Laplace transform, it is convenient to break the expressions for $y_1(s)$ and $y_2(s)$ into partial fractions. If the mass-damper-spring system is underdamped, then solutions of the form $e^{-\alpha t} \cos \omega t$ and $e^{-\alpha t} \sin \omega t$ will exist. The inverse Laplace transforms of these functions are:

$$e^{-\alpha t} \cos \omega t \Longleftrightarrow \frac{s + \alpha}{(s+\alpha)^2 + \omega^2} \tag{A18}$$

$$\frac{e^{-\alpha t} \sin \omega t}{\omega} \Longleftrightarrow \frac{1}{(s+\alpha)^2 + \omega^2} \tag{A19}$$

Hence, it is necessary to factor the fourth order denominator into two quadratic terms. This was performed numerically with a computer routine for every case. Since all cases are underdamped the roots consist of two complex conjugate pairs:

$$-\alpha_1 \pm \omega_1 i \Longleftrightarrow (s + \alpha_1 - \omega_1)(s + \alpha_1 + \omega_1) = (s + \alpha_1)^2 + \omega_1^2 \tag{A20}$$

and

$$-\alpha_2 \pm \omega_2 \Leftrightarrow (s + \alpha_2 - \omega_2)(s + \alpha_2 + \omega_2) = (s + \alpha_2)^2 + \omega_2^2 \tag{A21}$$

Hence:

$$d_4\, y_1(s) = \frac{e_1(s+\alpha_1) + e_2}{(s+\alpha_1)^2 + \omega_1^2} + \frac{e_3(s+\alpha_2) + e_4}{(s+\alpha_2)^2 + \omega_2^2} \tag{A22}$$

$$d_4\, y_2(s) = \frac{f_1(s+\alpha_1) + f_2}{(s+\alpha_1)^2 + \omega_1^2} + \frac{f_3(s+\alpha_2) + f_4}{(s+\alpha_2)^2 + \omega_2^2} \tag{A23}$$

where e_1, e_2, e_3, e_4, f_1, f_2, f_3, f_4 are determined by letting s approach $-\alpha_1 \pm \omega_1 i$ and $-\alpha_2 \pm \omega_2 i$. This is equivalent to defining Z_1, Z_2, Z_3 and Z_4 as:

$$Z_1 = \frac{a_3 s^3 + a_2 s^2 + a_1 s + a_0}{(s+\alpha_2)^2 + {}_2^2} \quad \text{with} \quad s = -\alpha_1 + \omega_1 i \tag{A24}$$

$$Z_2 = \frac{a_3 s^3 + a_2 s^2 + a_1 s + a_0}{(s+\alpha_1)^2 + \omega_1^2} \quad \text{with} \quad s = \alpha_2 \pm \omega_2 i \tag{A25}$$

$$Z_3 = \frac{b_3 s^3 + b_2 s^2 + b_1 s + b_0}{(s+\alpha_2)^2 + \omega_2^2} \quad \text{with} \quad s \quad \alpha_1 \pm \omega_1 i \tag{A26}$$

$$Z_4 = \frac{b_3 s^3 + b_2 s^2 + b_1 s + b_0}{(s+\alpha_1)^2 + {}_1^2} \qquad \text{with} \quad s = \alpha_2 \pm \omega_s i \tag{A27}$$

and then:

$$e_1 = \text{imag } Z_1/\omega_1 \qquad e_2 = \text{real } Z_1 \tag{A28}$$

$$e_3 = \text{imag } Z_2/\omega_2 \qquad e_4 = \text{real } Z_2 \tag{A29}$$

$$f_1 = \text{imag } Z_3/\omega_1 \qquad f_2 = \text{real } Z_3 \tag{A30}$$

$$f_3 = \text{imag } Z_4/\omega_2 \qquad f_4 = \text{real } Z_4 \tag{A31}$$

Finally, $y_1(s)$ and $y_2(s)$ can be "inverted", leading to the expressions:

$$y_1(t) = e^{-\alpha_1 t} \left[\frac{e_1}{d_4} \cos \omega_1 t + \frac{e_2}{d_{4\ 1}} \sin \omega_1 t \right] +$$

$$+ e^{-\alpha_2 t} \left[\frac{e_3}{d_4} \cos \omega_2 t + \frac{e_4}{d_{4\ 1}} \sin \omega_2 t \right] \tag{A32}$$

and

$$y_2(t) = e^{-\alpha_1 t}\left[\frac{f_1}{d_4}\cos\omega_1 t + \frac{f_2}{d_4\omega_1}\sin\omega_1 t\right]$$

$$+ e^{-\alpha_2 t}\left[\frac{f_3}{d_4}\cos\omega_2 t + \frac{f_4}{d_4\omega_2}\sin\omega_2 t\right] \qquad \text{(A33)}$$

These expressions can be written in the more convenient form:

$$y_1(t) = A_1 e^{-\alpha_1 t}\cos(\omega_1 t + \phi_1) + A_2 e^{-\alpha_2 t}\cos(\omega_2 t + \phi_2) \qquad \text{(A34)}$$

$$y_2(t) = A_3 e^{-\alpha_1 t}\cos(\omega_1 t + \phi_3) + A_4 e^{-\alpha_2 t}\cos(\omega_2 t + \phi_4) \qquad \text{(A35)}$$

where :

$$A_1 = [e_1{}^2 + (e_2/\omega_1)^2]^{\frac{1}{2}}/d_4 \qquad \text{(A36)}$$

$$A_2 = [e_3{}^2 + (e_4/\omega_2)^2]^{\frac{1}{2}}/d_4 \qquad \text{(A37)}$$

$$A_3 = [f_1{}^2 + (f_2/\omega_1)^2]^{\frac{1}{2}}/d_4 \qquad \text{(A38)}$$

$$A_4 = [f_3{}^2 + (f_3/\omega_2)^2]^{\frac{1}{2}}/d_4 \qquad \text{(A39)}$$

$$\phi_1 = -\tan^{-1} (e_2/e_1\omega_2) \tag{A40}$$

$$\phi_2 = -\tan^{-1} (e_4/e_3\omega_2) \tag{A41}$$

$$\phi_3 = -\tan^{-1} (f_2/f_1\omega_1) \tag{A42}$$

$$\phi_4 = -\tan^{-1} (f_4/f_3\omega_2) \tag{A43}$$

It is possible to choose m_1, c_1, k_1, m_2, c_2 and k_2 to obtain various desired values of α_1, α_2, ω_1 and ω_2 . The required relationships are simplified if the product m_1m_2 is arbitrarily set to 1. Then the following four simultaneous nonlinear equations are obtained:

$$c_1 + 2c_2 = 2(\alpha_1 + \alpha_2) \tag{A44}$$

$$k_1 + 2k_2 + c_1c_2 = \alpha_1^2 + \omega_1^2 + 4\alpha_1\alpha_2 + \alpha_2^2 + \omega_2^2 \tag{A45}$$

$$c_1k_2 + c_2k_1 = 2[\alpha_1(\alpha_2^2 + \omega_2^2) + \alpha_2(\alpha_1^2 + \omega_1^2)] \tag{A46}$$

$$k_1k_2 = (\alpha_1^2 + \omega_1^2) (\alpha_2^2 + \omega_2^2) \tag{A47}$$

It is possible to solve these numerically for c_1, c_2, k_1 and k_2 (see Reference [19]. That is, let $\underline{y}(\tau)$ be functions such that:

$$\underline{y}(\tau) = \begin{bmatrix} y_1(\tau) \\ y_2(\tau) \\ y_3(\tau) \\ y_4(\tau) \end{bmatrix} \tag{A48}$$

and let

$$\underline{y}(0) = \begin{bmatrix} 2\alpha_1 \\ \alpha_1^2 + \omega_1^2 \\ 2\alpha_2 \\ \alpha_2^2 + \omega_2^2 \end{bmatrix} \tag{A49}$$

Further, let

$$\underline{F}(\underline{y}) = \begin{bmatrix} y_1 + 2y_3 \\ y_2 + 2y_4 + y_1y_3 \\ y_1y_4 + y_3y_2 \\ y_2y_4 \end{bmatrix} - \begin{bmatrix} y_1(0) + y_3(0) \\ y_2(0) + y_1(0)y_3(0) + y_4(0) \\ y_1(0)y_4(0) + y_3(0)y_2(0) \\ y_2(0)y_4(0) \end{bmatrix} \tag{A50}$$

Thus

$$\underline{F}(0) = \begin{bmatrix} y_3(0) \\ y_4(0) \\ 0 \\ 0 \end{bmatrix} \tag{A51}$$

and

$$\underline{\nabla F} = \begin{bmatrix} 1 & 0 & 2 & 0 \\ y_3 & 1 & y_1 & 2 \\ y_4 & y_3 & y_2 & y_1 \\ 0 & y_4 & 0 & y_2 \end{bmatrix} \tag{A52}$$

Finally, let

$$\underline{F}(\tau) = (1 - \tau)\underline{F}(0) \tag{A53}$$

Then

$$F(1) = 0 \tag{A54}$$

and

$$\dot{\underline{F}}(\tau) = \underline{\nabla F}\dot{\underline{y}} = - \underline{F}(0) \tag{A55}$$

Hence,

$$\dot{\underline{y}} = - [\nabla F]^{-1}F(0) \tag{A56}$$

Equations (A56) form a system of four first order differential equations whose solution $y(\tau)$ evaluated at $\tau = 1$, provide the desired values of c_1, k_1, c_2, and k_2.

APPENDIX B

Solver Subroutine Listings

```
      SUBROUTINE DRK45(X,XDOT,T,H,M)
      IMPLICIT REAL*8 (A-H,O-Z)
C *** RUNGE-KUTTA OPTIMAL COEFFS ROUTINE.   TRUNCATION ERROR = H**6.
C        H = STEP SIZE
C        M = NUMBER OF FIRST ORDER DIFFERENTIAL EQUATIONS
C        T = INDEPENDENT VARIABLE
C        X IS THE STATE VECTOR
C        XDOT IS THE DERIVATIVE OF THE STATE VECTOR X.
C        THE VECTORS X(T+H) AND XDOT(T+H) ARE RETURNED
C        RK45 CALLS XDOTEQ(X,XDOT,T) WICH IS THE SUBROUTINE IN
C           WHICH THE M FIRST ORDER EQUATIONS ARE INPUT..
C        XDOTEQ INPUT IS AS FOLLOWS
C                    XDOT(1) = F(X,T)
C                    XDOT(2) = G(X,T)
C                              ETC.
C
      DIMENSION X( M),XDOT( M),XNEW(20),A(6),F(6,20),SAVE(20),B(6,5)
      DIMENSION XSAVE(20)
      DATA KOUNT/0/
      IF(KOUNT.GT.0) GO TO 5
      KOUNT=1
      A(1)=0.0D0
      A(2)=.2D0
      A(3)=.3D0
      A(4)=.6D0
      A(5)=1.D0
      A(6)=11.D0/12.D0
      C1=59.D0/594.D0
      C3=2750.D0/6993.D0
      C4=125.D0/513.D0
      C5=-1.D0/14.D0
      C6=2592.D0/7733.D0
      B(2,1)=.2D0
      B(3,1)=3.D0/40.D0
      B(3,2)=9.D0/40.D0
      B(4,1)=.3D0
      B(4,2)=-.9D0
      B(4,3)=1.2D0
      B(5,1)=-11.D0/54.D0
      B(5,2)=2.5D0
      B(5,3)=-70.D0/27.D0
      B(5,4)=35.D0/27.D0
      B(6,1)=-473.D0/10368.D0
      B(6,2)=1595.D0/1728.D0
      B(6,3)=-34595.D0/54432.D0
```

```
      B(6,4)=38665.D0/62208.D0
      B(6,5)=7733.D0/145152.D0
    5 DO 4 J=1,M
      SAVE(J)=0.0D0
    4 XSAVE(J)=X(J)
      DO 10 K=1,6
      IF(K.EQ.1) GO TO 30
      N=K-1
      DO 20 J=1,M
      SAVE(J)=0.0D0
      DO 40 LAM=1,N
   40 SAVE(J)=SAVE(J)+B(K,LAM)*F(LAM,J)
   20 SAVE(J)=H*SAVE(J)
   30 DO 50 N=1,M
   50 XNEW(N)=XSAVE(N)+SAVE(N)
      TNEW=T+A(K)*H
      CALL XDOTEQ(XNEW,XDOT,TNEW)
      DO 60 J=1,M
   60 F(K,J)=XDOT(J)
   10 CONTINUE
      DO 70 J=1,M
   70 X(J)=XSAVE(J)+H*(C1*F(1,J)+C3*F(3,J)+C4*F(4,J)+C5*F(5,J)+
     Q    C6*F(6,J))
      T=T+H
      RETURN
      END
```

```
      SUBROUTINE DRKGS(PRMT,Y,DERY,NDIM,IHLF,FCT,OUTP,AUX)
C                                                                       DRKG   50
C        PURPOSE                                                        DRKG   60
C           TO SOLVE A SYSTEM OF FIRST ORDER ORDINARY DIFFERENTIAL      DRKG   70
C           EQUATIONS WITH GIVEN INITIAL VALUES.                        DRKG   80
C                                                                       DRKG   90
C        USAGE                                                          DRKG  100
C           CALL DRKGS (PRMT,Y,DERY,NDIM,IHLF,FCT,OUTP,AUX)             DRKG  110
C           PARAMETERS FCT AND OUTP REQUIRE AN EXTERNAL STATEMENT.      DRKG  120
C                                                                       DRKG  130
C        DESCRIPTION OF PARAMETERS                                      DRKG  140
C           PRMT   - DOUBLE PRECISION INPUT AND OUTPUT VECTOR WITH      DRKG  150
C                    DIMENSION GREATER THAN OR EQUAL TO 5, WHICH        DRKG  160
C                    SPECIFIES THE PARAMETERS OF THE INTERVAL AND OF    DRKG  170
C                    ACCURACY AND WHICH SERVES FOR COMMUNICATION BETWEENDRKG  180
C                    OUTPUT SUBROUTINE (FURNISHED BY THE USER) AND      DRKG  190
C                    SUBROUTINE DRKGS. EXCEPT PRMT(5) THE COMPONENTS    DRKG  200
C                    ARE NOT DESTROYED BY SUBROUTINE DRKGS AND THEY ARE DRKG  210
C           PRMT(1)- LOWER BOUND OF THE INTERVAL (INPUT),               DRKG  220
C           PRMT(2)- UPPER BOUND OF THE INTERVAL (INPUT),               DRKG  230
C           PRMT(3)- INITIAL INCREMENT OF THE INDEPENDENT VARIABLE      DRKG  240
C                    (INPUT),                                           DRKG  250
C           PRMT(4)- UPPER ERROR BOUND (INPUT). IF ABSOLUTE ERROR IS    DRKG  260
C                    GREATER THAN PRMT(4), INCREMENT GETS HALVED.       DRKG  270
C                    IF INCREMENT IS LESS THAN PRMT(3) AND ABSOLUTE     DRKG  280
C                    ERROR LESS THAN PRMT(4)/50, INCREMENT GETS DOUBLED.DRKG  290
C                    THE USER MAY CHANGE PRMT(4) BY MEANS OF HIS        DRKG  300
C                    OUTPUT SUBROUTINE.                                 DRKG  310
C           PRMT(5)- NO INPUT PARAMETER. SUBROUTINE DRKGS INITIALIZES   DRKG  320
C                    PRMT(5)=0. IF THE USER WANTS TO TERMINATE          DRKG  330
C                    SUBROUTINE DRKGS AT ANY OUTPUT POINT, HE HAS TO    DRKG  340
C                    CHANGE PRMT(5) TO NON-ZERO BY MEANS OF SUBROUTINE  DRKG  350
C                    OUTP. FURTHER COMPONENTS OF VECTOR PRMT ARE        DRKG  360
C                    FEASIBLE IF ITS DIMENSION IS DEFINED GREATER       DRKG  370
C                    THAN 5. HOWEVER SUBROUTINE DRKGS DOES NOT REQUIRE  DRKG  380
C                    AND CHANGE THEM. NEVERTHELESS THEY MAY BE USEFUL   DRKG  390
C                    FOR HANDING RESULT VALUES TO THE MAIN PROGRAM      DRKG  400
C                    (CALLING DRKGS) WHICH ARE OBTAINED BY SPECIAL      DRKG  410
C                    MANIPULATIONS WITH OUTPUT DATA IN SUBROUTINE OUTP. DRKG  420
C           Y      - DOUBLE PRECISION INPUT VECTOR OF INITIAL VALUES    DRKG  430
C                    (DESTROYED). LATERON Y IS THE RESULTING VECTOR OF  DRKG  440
C                    DEPENDENT VARIABLES COMPUTED AT INTERMEDIATE       DRKG  450
C                    POINTS X.                                          DRKG  460
C           DERY   - DOUBLE PRECISION INPUT VECTOR OF ERROR WEIGHTS     DRKG  470
C                    (DESTROYED). THE SUM OF ITS COMPONENTS MUST BE     DRKG  480
```

```
C                EQUAL TO 1. LATERON DERY IS THE VECTOR OF            DRKG 490
C                DERIVATIVES, WHICH BELONG TO FUNCTION VALUES Y AT    DRKG 500
C                INTERMEDIATE POINTS X.                               DRKG 510
C        NDIM  - AN INPUT VALUE, WHICH SPECIFIES THE NUMBER OF        DRKG 520
C                EQUATIONS IN THE SYSTEM.                             DRKG 530
C        IHLF  - AN OUTPUT VALUE, WHICH SPECIFIES THE NUMBER OF       DRKG 540
C                BISECTIONS OF THE INITIAL INCREMENT. IF IHLF GETS    DRKG 550
C                GREATER THAN 10, SUBROUTINE DRKGS RETURNS WITH       DRKG 560
C                ERROR MESSAGE IHLF=11 INTO MAIN PROGRAM. ERROR       DRKG 570
C                MESSAGE IHLF=12 OR IHLF=13 APPEARS IN CASE           DRKG 580
C                PRMT(3)=0 OR IN CASE SIGN(PRMT(3)).NE.SIGN(PRMT(2)-DRKG 590
C                PRMT(1)) RESPECTIVELY.                               DRKG 600
C        FCT   - THE NAME OF AN EXTERNAL SUBROUTINE USED. THIS        DRKG 610
C                SUBROUTINE COMPUTES THE RIGHT HAND SIDES DERY OF     DRKG 620
C                THE SYSTEM TO GIVEN VALUES X AND Y. ITS PARAMETER    DRKG 630
C                LIST MUST BE X,Y,DERY. SUBROUTINE FCT SHOULD         DRKG 640
C                NOT DESTROY X AND Y.                                 DRKG 650
C        OUTP  - THE NAME OF AN EXTERNAL OUTPUT SUBROUTINE USED.      DRKG 660
C                ITS PARAMETER LIST MUST BE X,Y,DERY,IHLF,NDIM,PRMT.DRKG 670
C                NONE OF THESE PARAMETERS (EXCEPT, IF NECESSARY,      DRKG 680
C                PRMT(4),PRMT(5),...) SHOULD BE CHANGED BY            DRKG 690
C                SUBROUTINE OUTP. IF PRMT(5) IS CHANGED TO NON-ZERO,DRKG 700
C                SUBROUTINE DRKGS IS TERMINATED.                      DRKG 710
C        AUX   - DOUBLE PRECISION AUXILIARY STORAGE ARRAY WITH 8      DRKG 720
C                ROWS AND NDIM COLUMNS.                               DRKG 730
C                                                                     DRKG 740
C     REMARKS                                                         DRKG 750
C        THE PROCEDURE TERMINATES AND RETURNS TO CALLING PROGRAM, IF  DRKG 760
C        (1) MORE THAN 10 BISECTIONS OF THE INITIAL INCREMENT ARE     DRKG 770
C            NECESSARY TO GET SATISFACTORY ACCURACY (ERROR MESSAGE    DRKG 780
C            IHLF=11),                                                DRKG 790
C        (2) INITIAL INCREMENT IS EQUAL TO 0 OR HAS WRONG SIGN        DRKG 800
C            (ERROR MESSAGES IHLF=12 OR IHLF=13),                     DRKG 810
C        (3) THE WHOLE INTEGRATION INTERVAL IS WORKED THROUGH,        DRKG 820
C        (4) SUBROUTINE OUTP HAS CHANGED PRMT(5) TO NON-ZERO.         DRKG 830
C                                                                     DRKG 840
C     SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED                   DRKG 850
C        THE EXTERNAL SUBROUTINES FCT(X,Y,DERY) AND                   DRKG 860
C        OUTP(X,Y,DERY,IHLF,NDIM,PRMT) MUST BE FURNISHED BY THE USER.DRKG 870
C                                                                     DRKG 880
C     METHOD                                                          DRKG 890
C        EVALUATION IS DONE BY MEANS OF FOURTH ORDER RUNGE-KUTTA      DRKG 900
C        FORMULAE IN THE MODIFICATION DUE TO GILL. ACCURACY IS        DRKG 910
C        TESTED COMPARING THE RESULTS OF THE PROCEDURE WITH SINGLE    DRKG 920
C        AND DOUBLE INCREMENT.                                        DRKG 930
```

```
C           SUBROUTINE DRKGS AUTOMATICALLY ADJUSTS THE INCREMENT DURING  DRKG 940
C           THE WHOLE COMPUTATION BY HALVING OR DOUBLING. IF MORE THAN   DRKG 950
C           10 BISECTIONS OF THE INCREMENT ARE NECESSARY TO GET          DRKG 960
C           SATISFACTORY ACCURACY, THE SUBROUTINE RETURNS WITH           DRKG 970
C           ERROR MESSAGE IHLF=11 INTO MAIN PROGRAM.                     DRKG 980
C           TO GET FULL FLEXIBILITY IN OUTPUT, AN OUTPUT SUBROUTINE      DRKG 990
C           MUST BE FURNISHED BY THE USER.                               DRKG1000
C           FOR REFERENCE, SEE                                           DRKG1010
C           RALSTON/WILF, MATHEMATICAL METHODS FOR DIGITAL COMPUTERS,     DRKG1020
C           WILEY, NEW YORK/LONDON, 1960, PP.110-120.                    DRKG1030
C                                                                        DRKG1040
C     ..................................................................DRKG1050
C                                                                        DRKG1060
C                                                                        DRKG1080
C                                                                        DRKG1090
      DIMENSION Y(NDIM),DERY(NDIM),AUX(8,NDIM),A(4),B(4),C(4),PRMT(5)    DRKG1100
      DOUBLE PRECISION PRMT,Y,DERY,AUX,A,B,C,X,XEND,H,AJ,BJ,CJ,R1,R2,    DRKG1110
     1DELT,DABS                                                          DRKG1120
      DO 1 I=1,NDIM                                                      DRKG1130
    1 AUX(8,I)=.06666666666666667 D0*DERY(I)                             DRKG1140
      X=PRMT(1)                                                          DRKG1150
      XEND=PRMT(2)                                                       DRKG1160
      H=PRMT(3)                                                          DRKG1170
      PRMT(5)=0.D0                                                       DRKG1180
      CALL FCT(X,Y,DERY)                                                 DRKG1190
C                                                                        DRKG1200
C     ERROR TEST                                                         DRKG1210
      IF(H*(XEND-X))38,37,2                                              DRKG1220
C                                                                        DRKG1230
C     PREPARATIONS FOR RUNGE-KUTTA METHOD                                DRKG1240
    2 A(1)=.5D0                                                          DRKG1250
      A(2)=.2928932188134525 D0                                          DRKG1260
      A(3)=1.707106781186548 D0                                          DRKG1270
      A(4)=.1666666666666667 D0                                          DRKG1280
      B(1)=2.D0                                                          DRKG1290
      B(2)=1.D0                                                          DRKG1300
      B(3)=1.D0                                                          DRKG1310
      B(4)=2.D0                                                          DRKG1320
      C(1)=.5D0                                                          DRKG1330
      C(2)=.2928932188134525 D0                                          DRKG1340
      C(3)=1.707106781186548 D0                                          DRKG1350
      C(4)=.5D0                                                          DRKG1360
C                                                                        DRKG1370
C     PREPARATIONS OF FIRST RUNGE-KUTTA STEP                             DRKG1380
      DO 3 I=1,NDIM                                                      DRKG1390
```

```
      AUX(1,I)=Y(I)                                                     DRKG1400
      AUX(2,I)=DERY(I)                                                  DRKG1410
      AUX(3,I)=0.D0                                                     DRKG1420
    3 AUX(6,I)=0.D0                                                     DRKG1430
      IREC=0                                                            DRKG1440
      H=H+H                                                             DRKG1450
      IHLF=-1                                                           DRKG1460
      ISTEP=0                                                           DRKG1470
      IEND=0                                                            DRKG1480
C                                                                       DRKG1490
C                                                                       DRKG1500
C     START OF A RUNGE-KUTTA STEP                                       DRKG1510
    4 IF((X+H-XEND)*H)7,6,5                                             DRKG1520
    5 H=XEND-X                                                          DRKG1530
    6 IEND=1                                                            DRKG1540
C                                                                       DRKG1550
C     RECORDING OF INITIAL VALUES OF THIS STEP                          DRKG1560
    7 CALL OUTP(X,Y,DERY,IREC,NDIM,PRMT)                                DRKG1570
      IF(PRMT(5))40,8,40                                                DRKG1580
    8 ITEST=0                                                           DRKG1590
    9 ISTEP=ISTEP+1                                                     DRKG1600
C                                                                       DRKG1610
C                                                                       DRKG1620
C     START OF INNERMOST RUNGE-KUTTA LOOP                               DRKG1630
      J=1                                                               DRKG1640
   10 AJ=A(J)                                                           DRKG1650
      BJ=B(J)                                                           DRKG1660
      CJ=C(J)                                                           DRKG1670
      DO 11 I=1,NDIM                                                    DRKG1680
      R1=H*DERY(I)                                                      DRKG1690
      R2=AJ*(R1-BJ*AUX(6,I))                                            DRKG1700
      Y(I)=Y(I)+R2                                                      DRKG1710
      R2=R2+R2+R2                                                       DRKG1720
   11 AUX(6,I)=AUX(6,I)+R2-CJ*R1                                        DRKG1730
      IF(J-4)12,15,15                                                   DRKG1740
   12 J=J+1                                                             DRKG1750
      IF(J-3)13,14,13                                                   DRKG1760
   13 X=X+.5D0*H                                                        DRKG1770
   14 CALL FCT(X,Y,DERY)                                                DRKG1780
      GOTO 10                                                           DRKG1790
C     END OF INNERMOST RUNGE-KUTTA LOOP                                 DRKG1800
C                                                                       DRKG1810
C                                                                       DRKG1820
C     TEST OF ACCURACY                                                  DRKG1830
   15 IF(ITEST)16,16,20                                                 DRKG1840
```

```
C                                                                               DRKG1850
C     IN CASE ITEST=0 THERE IS NO POSSIBILITY FOR TESTING OF ACCURACY          DRKG1860
   16 DO 17 I=1,NDIM                                                            DRKG1870
   17 AUX(4,I)=Y(I)                                                             DRKG1880
      ITEST=1                                                                   DRKG1890
      ISTEP=ISTEP+ISTEP-2                                                       DRKG1900
   18 IHLF=IHLF+1                                                               DRKG1910
      X=X-H                                                                     DRKG1920
      H=.5D0*H                                                                  DRKG1930
      DO 19 I=1,NDIM                                                            DRKG1940
      Y(I)=AUX(1,I)                                                             DRKG1950
      DERY(I)=AUX(2,I)                                                          DRKG1960
   19 AUX(6,I)=AUX(3,I)                                                         DRKG1970
      GOTO 9                                                                    DRKG1980
C                                                                               DRKG1990
C     IN CASE ITEST=1 TESTING OF ACCURACY IS POSSIBLE                           DRKG2000
   20 IMOD=ISTEP/2                                                              DRKG2010
      IF(ISTEP-IMOD-IMOD)21,23,21                                               DRKG2020
   21 CALL FCT(X,Y,DERY)                                                        DRKG2030
      DO 22 I=1,NDIM                                                            DRKG2040
      AUX(5,I)=Y(I)                                                             DRKG2050
   22 AUX(7,I)=DERY(I)                                                          DRKG2060
      GOTO 9                                                                    DRKG2070
C                                                                               DRKG2080
C     COMPUTATION OF TEST VALUE DELT                                            DRKG2090
   23 DELT=0.D0                                                                 DRKG2100
      DO 24 I=1,NDIM                                                            DRKG2110
   24 DELT=DELT+AUX(8,I)*DABS(AUX(4,I)-Y(I))                                    DRKG2120
      IF(DELT-PRMT(4))28,28,25                                                  DRKG2130
C                                                                               DRKG2140
C     ERROR IS TOO GREAT                                                        DRKG2150
   25 IF(IHLF-30)26,36,36                                                       DRKG2160
   26 DO 27 I=1,NDIM                                                            DRKG2170
   27 AUX(4,I)=AUX(5,I)                                                         DRKG2180
      ISTEP=ISTEP+ISTEP-4                                                       DRKG2190
      X=X-H                                                                     DRKG2200
      IEND=0                                                                    DRKG2210
      GOTO 18                                                                   DRKG2220
C                                                                               DRKG2230
C     RESULT VALUES ARE GOOD                                                    DRKG2240
   28 CALL FCT(X,Y,DERY)                                                        DRKG2250
      DO 29 I=1,NDIM                                                            DRKG2260
      AUX(1,I)=Y(I)                                                             DRKG2270
      AUX(2,I)=DERY(I)                                                          DRKG2280
      AUX(3,I)=AUX(6,I)                                                         DRKG2290
```

```
      Y(I)=AUX(5,I)                                                     DRKG2300
   29 DERY(I)=AUX(7,I)                                                  DRKG2310
      CALL OUTP(X-H,Y,DERY,IHLF,NDIM,PRMT)                              DRKG2320
      IF(PRMT(5))40,30,40                                               DRKG2330
   30 DO 31 I=1,NDIM                                                    DRKG2340
      Y(I)=AUX(1,I)                                                     DRKG2350
   31 DERY(I)=AUX(2,I)                                                  DRKG2360
      IREC=IHLF                                                         DRKG2370
      IF(IEND)32,32,39                                                  DRKG2380
C                                                                       DRKG2390
C     INCREMENT GETS DOUBLED                                            DRKG2400
   32 IHLF=IHLF-1                                                       DRKG2410
      ISTEP=ISTEP/2                                                     DRKG2420
      H=H+H                                                             DRKG2430
      IF(IHLF)4,33,33                                                   DRKG2440
   33 IMOD=ISTEP/2                                                      DRKG2450
      IF(ISTEP-IMOD-IMOD)4,34,4                                         DRKG2460
   34 IF(DELT-.02D0*PRMT(4))35,35,4                                     DRKG2470
   35 IHLF=IHLF-1                                                       DRKG2480
      ISTEP=ISTEP/2                                                     DRKG2490
      H=H+H                                                             DRKG2500
      GOTO 4                                                            DRKG2510
C                                                                       DRKG2520
C                                                                       DRKG2530
C     RETURNS TO CALLING PROGRAM                                        DRKG2540
   36 IHLF=31                                                           DRKG2550
      CALL FCT(X,Y,DERY)                                                DRKG2560
      GOTO 39                                                           DRKG2570
   37 IHLF=32                                                           DRKG2580
      GOTO 39                                                           DRKG2590
   38 IHLF=33                                                           DRKG2600
   39 CALL OUTP(X,Y,DERY,IHLF,NDIM,PRMT)                                DRKG2610
   40 RETURN                                                            DRKG2620
      END                                                               DRKG2630
```

```
      SUBROUTINE DHPCG(PRMT,Y,DERY,NDIM,IHLF,FCT,OUTP,AUX)
C                                                                        DHCG   50
C        PURPOSE                                                         DHCG   60
C           TO SOLVE A SYSTEM OF FIRST ORDER ORDINARY GENERAL            DHCG   70
C           DIFFERENTIAL EQUATIONS WITH GIVEN INITIAL VALUES.            DHCG   80
C                                                                        DHCG   90
C        USAGE                                                           DHCG  100
C           CALL DHPCG (PRMT,Y,DERY,NDIM,IHLF,FCT,OUTP,AUX)              DHCG  110
C           PARAMETERS FCT AND OUTP REQUIRE AN EXTERNAL STATEMENT.       DHCG  120
C                                                                        DHCG  130
C        DESCRIPTION OF PARAMETERS                                       DHCG  140
C           PRMT   - DOUBLE PRECISION INPUT AND OUTPUT VECTOR WITH       DHCG  150
C                    DIMENSION GREATER THAN OR EQUAL TO 5, WHICH         DHCG  160
C                    SPECIFIES THE PARAMETERS OF THE INTERVAL AND OF     DHCG  170
C                    ACCURACY AND WHICH SERVES FOR COMMUNICATION BETWEENDHCG  180
C                    OUTPUT SUBROUTINE (FURNISHED BY THE USER) AND       DHCG  190
C                    SUBROUTINE DHPCG. EXCEPT PRMT(5) THE COMPONENTS     DHCG  200
C                    ARE NOT DESTROYED BY SUBROUTINE DHPCG AND THEY ARE  DHCG  210
C           PRMT(1)- LOWER BOUND OF THE INTERVAL (INPUT),                DHCG  220
C           PRMT(2)- UPPER BOUND OF THE INTERVAL (INPUT),                DHCG  230
C           PRMT(3)- INITIAL INCREMENT OF THE INDEPENDENT VARIABLE       DHCG  240
C                    (INPUT),                                            DHCG  250
C           PRMT(4)- UPPER ERROR BOUND (INPUT). IF ABSOLUTE ERROR IS     DHCG  260
C                    GREATER THAN PRMT(4), INCREMENT GETS HALVED.        DHCG  270
C                    IF INCREMENT IS LESS THAN PRMT(3) AND ABSOLUTE      DHCG  280
C                    ERROR LESS THAN PRMT(4)/50, INCREMENT GETS DOUBLED.DHCG  290
C                    THE USER MAY CHANGE PRMT(4) BY MEANS OF HIS         DHCG  300
C                    OUTPUT SUBROUTINE.                                  DHCG  310
C           PRMT(5)- NO INPUT PARAMETER. SUBROUTINE DHPCG INITIALIZES    DHCG  320
C                    PRMT(5)=0. IF THE USER WANTS TO TERMINATE           DHCG  330
C                    SUBROUTINE DHPCG AT ANY OUTPUT POINT, HE HAS TO     DHCG  340
C                    CHANGE PRMT(5) TO NON-ZERO BY MEANS OF SUBROUTINE   DHCG  350
C                    OUTP. FURTHER COMPONENTS OF VECTOR PRMT ARE         DHCG  360
C                    FEASIBLE IF ITS DIMENSION IS DEFINED GREATER        DHCG  370
C                    THAN 5. HOWEVER SUBROUTINE DHPCG DOES NOT REQUIRE   DHCG  380
C                    AND CHANGE THEM. NEVERTHELESS THEY MAY BE USEFUL    DHCG  390
C                    FOR HANDING RESULT VALUES TO THE MAIN PROGRAM       DHCG  400
C                    (CALLING DHPCG) WHICH ARE OBTAINED BY SPECIAL       DHCG  410
C                    MANIPULATIONS WITH OUTPUT DATA IN SUBROUTINE OUTP.  DHCG  420
C           Y      - DOUBLE PRECISION INPUT VECTOR OF INITIAL VALUES     DHCG  430
C                    (DESTROYED). LATERON Y IS THE RESULTING VECTOR OF   DHCG  440
C                    DEPENDENT VARIABLES COMPUTED AT INTERMEDIATE        DHCG  450
C                    POINTS X.                                           DHCG  460
C           DERY   - DOUBLE PRECISION INPUT VECTOR OF ERROR WEIGHTS      DHCG  470
C                    (DESTROYED). THE SUM OF ITS COMPONENTS MUST BE      DHCG  480
```

```
C              EQUAL TO 1. LATERON DERY IS THE VECTOR OF             DHCG 490
C              DERIVATIVES, WHICH BELONG TO FUNCTION VALUES Y AT     DHCG 500
C              INTERMEDIATE POINTS X.                                DHCG 510
C     NDIM   - AN INPUT VALUE, WHICH SPECIFIES THE NUMBER OF         DHCG 520
C              EQUATIONS IN THE SYSTEM.                              DHCG 530
C     IHLF   - AN OUTPUT VALUE, WHICH SPECIFIES THE NUMBER OF        DHCG 540
C              BISECTIONS OF THE INITIAL INCREMENT. IF IHLF GETS     DHCG 550
C              GREATER THAN 10, SUBROUTINE DHPCG RETURNS WITH        DHCG 560
C              ERROR MESSAGE IHLF=11 INTO MAIN PROGRAM.              DHCG 570
C              ERROR MESSAGE IHLF=12 OR IHLF=13 APPEARS IN CASE      DHCG 580
C              PRMT(3)=0 OR IN CASE SIGN(PRMT(3)).NE.SIGN(PRMT(2)-DHCG 590
C              PRMT(1)) RESPECTIVELY.                                DHCG 600
C     FCT    - THE NAME OF AN EXTERNAL SUBROUTINE USED. IT           DHCG 610
C              COMPUTES THE RIGHT HAND SIDES DERY OF THE SYSTEM      DHCG 620
C              TO GIVEN VALUES OF X AND Y. ITS PARAMETER LIST        DHCG 630
C              MUST BE X,Y,DERY. THE SUBROUTINE SHOULD NOT           DHCG 640
C              DESTROY X AND Y.                                      DHCG 650
C     OUTP   - THE NAME OF AN EXTERNAL OUTPUT SUBROUTINE USED.       DHCG 660
C              ITS PARAMETER LIST MUST BE X,Y,DERY,IHLF,NDIM,PRMT.DHCG 670
C              NONE OF THESE PARAMETERS (EXCEPT, IF NECESSARY,       DHCG 680
C              PRMT(4),PRMT(5),...) SHOULD BE CHANGED BY             DHCG 690
C              SUBROUTINE OUTP. IF PRMT(5) IS CHANGED TO NON-ZERO,DHCG 700
C              SUBROUTINE DHPCG IS TERMINATED.                       DHCG 710
C     AUX    - DOUBLE PRECISION AUXILIARY STORAGE ARRAY WITH 16      DHCG 720
C              ROWS AND NDIM COLUMNS.                                DHCG 730
C                                                                    DHCG 740
C  REMARKS                                                           DHCG 750
C     THE PROCEDURE TERMINATES AND RETURNS TO CALLING PROGRAM, IF    DHCG 760
C     (1) MORE THAN 10 BISECTIONS OF THE INITIAL INCREMENT ARE       DHCG 770
C         NECESSARY TO GET SATISFACTORY ACCURACY (ERROR MESSAGE      DHCG 780
C         IHLF=11),                                                  DHCG 790
C     (2) INITIAL INCREMENT IS EQUAL TO 0 OR HAS WRONG SIGN          DHCG 800
C         (ERROR MESSAGES IHLF=12 OR IHLF=13),                       DHCG 810
C     (3) THE WHOLE INTEGRATION INTERVAL IS WORKED THROUGH,          DHCG 820
C     (4) SUBROUTINE OUTP HAS CHANGED PRMT(5) TO NON-ZERO.           DHCG 830
C                                                                    DHCG 840
C  SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED                     DHCG 850
C     THE EXTERNAL SUBROUTINES FCT(X,Y,DERY) AND                     DHCG 860
C     OUTP(X,Y,DERY,IHLF,NDIM,PRMT) MUST BE FURNISHED BY THE USER.DHCG 870
C                                                                    DHCG 880
C  METHOD                                                            DHCG 890
C     EVALUATION IS DONE BY MEANS OF HAMMINGS MODIFIED PREDICTOR-    DHCG 900
C     CORRECTOR METHOD. IT IS A FOURTH ORDER METHOD, USING 4         DHCG 910
C     PRECEEDING POINTS FOR COMPUTATION OF A NEW VECTOR Y OF THE     DHCG 920
C     DEPENDENT VARIABLES.                                           DHCG 930
```

```
C               FOURTH ORDER RUNGE-KUTTA METHOD SUGGESTED BY RALSTON IS      DHCG 940
C               USED FOR ADJUSTMENT OF THE INITIAL INCREMENT AND FOR          DHCG 950
C               COMPUTATION OF STARTING VALUES.                               DHCG 960
C               SUBROUTINE DHPCG AUTOMATICALLY ADJUSTS THE INCREMENT DURING   DHCG 970
C               THE WHOLE COMPUTATION BY HALVING OR DOUBLING.                 DHCG 980
C               TO GET FULL FLEXIBILITY IN OUTPUT, AN OUTPUT SUBROUTINE       DHCG 990
C               MUST BE CODED BY THE USER.                                    DHCG1000
C               FOR REFERENCE, SEE                                            DHCG1010
C               (1)  RALSTON/WILF, MATHEMATICAL METHODS FOR DIGITAL           DHCG1020
C                    COMPUTERS, WILEY, NEW YORK/LONDON, 1960, PP.95-109.      DHCG1030
C               (2)  RALSTON, RUNGE-KUTTA METHODS WITH MINIMUM ERROR BOUNDS,DHCG1040
C                    MTAC, VOL.16, ISS.80 (1962), PP.431-437.                 DHCG1050
C                                                                             DHCG1060
C     ..............................................................DHCG1070
C                                                                             DHCG1080
C                                                                             DHCG1100
C                                                                             DHCG1110
      DIMENSION PRMT(5),Y(NDIM),DERY(NDIM),AUX(16,NDIM)
      DOUBLE PRECISION Y,DERY,AUX,PRMT,X,H,Z,DELT,DABS
      N=1                                                                     DHCG1140
      IHLF=0                                                                  DHCG1150
      X=PRMT(1)                                                               DHCG1160
      H=PRMT(3)                                                               DHCG1170
      PRMT(5)=0.D0                                                            DHCG1180
      DO 1 I=1,NDIM                                                           DHCG1190
      AUX(16,I)=0.D0                                                          DHCG1200
      AUX(15,I)=DERY(I)                                                       DHCG1210
    1 AUX(1,I)=Y(I)                                                           DHCG1220
      IF(H*(PRMT(2)-X))3,2,4                                                  DHCG1230
C                                                                             DHCG1240
C     ERROR RETURNS                                                           DHCG1250
    2 IHLF=12                                                                 DHCG1260
      GOTO 4                                                                  DHCG1270
    3 IHLF=13                                                                 DHCG1280
C                                                                             DHCG1290
C     COMPUTATION OF DERY FOR STARTING VALUES                                 DHCG1300
    4 CALL FCT(X,Y,DERY)                                                      DHCG1310
C                                                                             DHCG1320
C     RECORDING OF STARTING VALUES                                            DHCG1330
      CALL OUTP(X,Y,DERY,IHLF,NDIM,PRMT)                                      DHCG1340
      IF(PRMT(5))6,5,6                                                        DHCG1350
    5 IF(IHLF)7,7,6                                                           DHCG1360
    6 RETURN                                                                  DHCG1370
    7 DO 8 I=1,NDIM                                                           DHCG1380
    8 AUX(8,I)=DERY(I)                                                        DHCG1390
```

```
C                                                                       DHCG1400
C        COMPUTATION OF AUX(2,1)                                        DHCG1410
         ISW=1                                                          DHCG1420
         GOTO 100                                                       DHCG1430
C                                                                       DHCG1440
       9 X=X+H                                                          DHCG1450
         DO 10 I=1,NDIM                                                 DHCG1460
      10 AUX(2,I)=Y(I)                                                  DHCG1470
C                                                                       DHCG1480
C        INCREMENT H IS TESTED BY MEANS OF BISECTION                    DHCG1490
      11 IHLF=IHLF+1                                                    DHCG1500
         X=X-H                                                          DHCG1510
         DO 12 I=1,NDIM                                                 DHCG1520
      12 AUX(4,I)=AUX(2,I)                                              DHCG1530
         H=.5D0*H                                                       DHCG1540
         N=1                                                            DHCG1550
         ISW=2                                                          DHCG1560
         GOTO 100                                                       DHCG1570
C                                                                       DHCG1580
      13 X=X+H                                                          DHCG1590
         CALL FCT(X,Y,DERY)                                             DHCG1600
         N=2                                                            DHCG1610
         DO 14 I=1,NDIM                                                 DHCG1620
         AUX(2,I)=Y(I)                                                  DHCG1630
      14 AUX(9,I)=DERY(I)                                               DHCG1640
         ISW=3                                                          DHCG1650
         GOTO 100                                                       DHCG1660
C                                                                       DHCG1670
C        COMPUTATION OF TEST VALUE DELT                                 DHCG1680
      15 DELT=0.D0                                                      DHCG1690
         DO 16 I=1,NDIM                                                 DHCG1700
      16 DELT=DELT+AUX(15,I)*DABS(Y(I)-AUX(4,I))                        DHCG1710
         DELT=.066666666666666667D0*DELT                                DHCG1720
         IF(DELT-PRMT(4))19,19,17                                       DHCG1730
      17 IF(IHLF-20)11,18,18                                            DHCG1740
C                                                                       DHCG1750
C        NO SATISFACTORY ACCURACY AFTER 10 BISECTIONS. ERROR MESSAGE.   DHCG1760
      18 IHLF=21                                                        DHCG1770
         X=X+H                                                          DHCG1780
         GOTO 4                                                         DHCG1790
C                                                                       DHCG1800
C        THERE IS SATISFACTORY ACCURACY AFTER LESS THAN 11 BISECTIONS.  DHCG1810
      19 X=X+H                                                          DHCG1820
         CALL FCT(X,Y,DERY)                                             DHCG1830
         DO 20 I=1,NDIM                                                 DHCG1840
```

```
      AUX(3,I)=Y(I)                                                     DHCG1850
   20 AUX(10,I)=DERY(I)                                                 DHCG1860
      N=3                                                               DHCG1870
      ISW=4                                                             DHCG1880
      GOTO 100                                                          DHCG1890
C                                                                       DHCG1900
   21 N=1                                                               DHCG1910
      X=X+H                                                             DHCG1920
      CALL FCT(X,Y,DERY)                                                DHCG1930
      X=PRMT(1)                                                         DHCG1940
      DO 22 I=1,NDIM                                                    DHCG1950
      AUX(11,I)=DERY(I)                                                 DHCG1960
   22 Y(I)=AUX(1,I)+H*(.375D0*AUX(8,I)+.7916666666666667D0*AUX(9,I)     DHCG1970
     1-.2083333333333333D0*AUX(10,I)+.0416666666666667D0*DERY(I))       DHCG1980
   23 X=X+H                                                             DHCG1990
      N=N+1                                                             DHCG2000
      CALL FCT(X,Y,DERY)                                                DHCG2010
      CALL OUTP(X,Y,DERY,IHLF,NDIM,PRMT)                                DHCG2020
      IF(PRMT(5))6,24,6                                                 DHCG2030
   24 IF(N-4)25,200,200                                                 DHCG2040
   25 DO 26 I=1,NDIM                                                    DHCG2050
      AUX(N,I)=Y(I)                                                     DHCG2060
   26 AUX(N+7,I)=DERY(I)                                                DHCG2070
      IF(N-3)27,29,200                                                  DHCG2080
C                                                                       DHCG2090
   27 DO 28 I=1,NDIM                                                    DHCG2100
      DELT=AUX(9,I)+AUX(9,I)                                            DHCG2110
      DELT=DELT+DELT                                                    DHCG2120
   28 Y(I)=AUX(1,I)+.3333333333333333D0*H*(AUX(8,I)+DELT+AUX(10,I))     DHCG2130
      GOTO 23                                                           DHCG2140
C                                                                       DHCG2150
   29 DO 30 I=1,NDIM                                                    DHCG2160
      DELT=AUX(9,I)+AUX(10,I)                                           DHCG2170
      DELT=DELT+DELT+DELT                                               DHCG2180
   30 Y(I)=AUX(1,I)+.375D0*H*(AUX(8,I)+DELT+AUX(11,I))                  DHCG2190
      GOTO 23                                                           DHCG2200
C                                                                       DHCG2210
C     THE FOLLOWING PART OF SUBROUTINE DHPCG COMPUTES BY MEANS OF       DHCG2220
C     RUNGE-KUTTA METHOD STARTING VALUES FOR THE NOT SELF-STARTING      DHCG2230
C     PREDICTOR-CORRECTOR METHOD.                                       DHCG2240
  100 DO 101 I=1,NDIM                                                   DHCG2250
      Z=H*AUX(N+7,I)                                                    DHCG2260
      AUX(5,I)=Z                                                        DHCG2270
  101 Y(I)=AUX(N,I)+.4D0*Z                                              DHCG2280
C     Z IS AN AUXILIARY STORAGE LOCATION                                DHCG2290
```

```
C                                                                         DHCG2300
      Z=X+.4D0*H                                                          DHCG2310
      CALL FCT(Z,Y,DERY)                                                  DHCG2320
      DO 102 I=1,NDIM                                                     DHCG2330
      Z=H*DERY(I)                                                         DHCG2340
      AUX(6,I)=Z                                                          DHCG2350
  102 Y(I)=AUX(N,I)+.29697760924775360D0*AUX(5,I)+.15875964497103583D0*ZDHCG2360
C                                                                         DHCG2370
      Z=X+.45573725421878943D0*H                                          DHCG2380
      CALL FCT(Z,Y,DERY)                                                  DHCG2390
      DO 103 I=1,NDIM                                                     DHCG2400
      Z=H*DERY(I)                                                         DHCG2410
      AUX(7,I)=Z                                                          DHCG2420
  103 Y(I)=AUX(N,I)+.21810038822592047D0*AUX(5,I)-3.05096514869293080D0* DHCG2430
     1AUX(6,I)+3.83286476046701030D0*Z                                    DHCG2440
C                                                                         DHCG2450
      Z=X+H                                                               DHCG2460
      CALL FCT(Z,Y,DERY)                                                  DHCG2470
      DO 104 I=1,NDIM                                                     DHCG2480
  104 Y(I)=AUX(N,I)+.17476028226269037D0*AUX(5,I)-.55148066287873294D0*  DHCG2490
     1AUX(6,I)+1.20553559939652350D0*AUX(7,I)+.17118478121951903D0*       DHCG2500
     2H*DERY(I)                                                           DHCG2510
      GOTO(9,13,15,21),ISW                                                DHCG2520
C                                                                         DHCG2530
C     POSSIBLE BREAK-POINT FOR LINKAGE                                    DHCG2540
C                                                                         DHCG2550
C     STARTING VALUES ARE COMPUTED.                                       DHCG2560
C     NOW START HAMMINGS MODIFIED PREDICTOR-CORRECTOR METHOD.             DHCG2570
  200 ISTEP=3                                                             DHCG2580
  201 IF(N-8)204,202,204                                                  DHCG2590
C                                                                         DHCG2600
C     N=8 CAUSES THE ROWS OF AUX TO CHANGE THEIR STORAGE LOCATIONS        DHCG2610
  202 DO 203 N=2,7                                                        DHCG2620
      DO 203 I=1,NDIM                                                     DHCG2630
      AUX(N-1,I)=AUX(N,I)                                                 DHCG2640
  203 AUX(N+6,I)=AUX(N+7,I)                                               DHCG2650
      N=7                                                                 DHCG2660
C                                                                         DHCG2670
C     N LESS THAN 8 CAUSES N+1 TO GET N                                   DHCG2680
  204 N=N+1                                                               DHCG2690
C                                                                         DHCG2700
C     COMPUTATION OF NEXT VECTOR Y                                        DHCG2710
      DO 205 I=1,NDIM                                                     DHCG2720
      AUX(N-1,I)=Y(I)                                                     DHCG2730
  205 AUX(N+6,I)=DERY(I)                                                  DHCG2740
```

```
      X=X+H                                                             DHCG2750
  206 ISTEP=ISTEP+1                                                     DHCG2760
      DO 207 I=1,NDIM                                                   DHCG2770
     0DELT=AUX(N-4,I)+1.3333333333333333D0*H*(AUX(N+6,I)+AUX(N+6,I)-    DHCG2780
     1AUX(N+5,I)+AUX(N+4,I)+AUX(N+4,I))                                 DHCG2790
      Y(I)=DELT-.9256198347107438D0*AUX(16,I)                           DHCG2800
  207 AUX(16,I)=DELT                                                    DHCG2810
C     PREDICTOR IS NOW GENERATED IN ROW 16 OF AUX, MODIFIED PREDICTOR   DHCG2820
C     IS GENERATED IN Y. DELT MEANS AN AUXILIARY STORAGE.               DHCG2830
C                                                                       DHCG2840
      CALL FCT(X,Y,DERY)                                                DHCG2850
C     DERIVATIVE OF MODIFIED PREDICTOR IS GENERATED IN DERY             DHCG2860
C                                                                       DHCG2870
      DO 208 I=1,NDIM                                                   DHCG2880
     0DELT=.125D0*(9.D0*AUX(N-1,I)-AUX(N-3,I)+3.D0*H*(DERY(I)+AUX(N+6,I)DHCG2890
     1+AUX(N+6,I)-AUX(N+5,I)))                                          DHCG2900
      AUX(16,I)=AUX(16,I)-DELT                                          DHCG2910
  208 Y(I)=DELT+.07438016528925620D0*AUX(16,I)                          DHCG2920
C                                                                       DHCG2930
C     TEST WHETHER H MUST BE HALVED OR DOUBLED                          DHCG2940
      DELT=0.D0                                                         DHCG2950
      DO 209 I=1,NDIM                                                   DHCG2960
  209 DELT=DELT+AUX(15,I)*DABS(AUX(16,I))                               DHCG2970
      IF(DELT-PRMT(4))210,222,222                                       DHCG2980
C                                                                       DHCG2990
C     H MUST NOT BE HALVED. THAT MEANS Y(I) ARE GOOD.                   DHCG3000
  210 CALL FCT(X,Y,DERY)                                                DHCG3010
      CALL OUTP(X,Y,DERY,IHLF,NDIM,PRMT)                                DHCG3020
      IF(PRMT(5))212,211,212                                            DHCG3030
  211 IF(IHLF-21)213,212,212                                            DHCG3040
  212 RETURN                                                            DHCG3050
  213 IF(H*(X-PRMT(2)))214,212,212                                      DHCG3060
  214 IF(DABS(X-PRMT(2))-.1D0*DABS(H))212,215,215                       DHCG3070
  215 IF(DELT-.02D0*PRMT(4))216,216,201                                 DHCG3080
C                                                                       DHCG3090
C                                                                       DHCG3100
C     H COULD BE DOUBLED IF ALL NECESSARY PRECEEDING VALUES ARE         DHCG3110
C     AVAILABLE                                                         DHCG3120
  216 IF(IHLF)201,201,217                                               DHCG3130
  217 IF(N-7)201,218,218                                                DHCG3140
  218 IF(ISTEP-4)201,219,219                                            DHCG3150
  219 IMOD=ISTEP/2                                                      DHCG3160
      IF(ISTEP-IMOD-IMOD)201,220,201                                    DHCG3170
  220 H=H+H                                                             DHCG3180
      IHLF=IHLF-1                                                       DHCG3190
```

```
      ISTEP=0                                                           DHCG3200
      DO 221 I=1,NDIM                                                   DHCG3210
      AUX(N-1,I)=AUX(N-2,I)                                             DHCG3220
      AUX(N-2,I)=AUX(N-4,I)                                             DHCG3230
      AUX(N-3,I)=AUX(N-6,I)                                             DHCG3240
      AUX(N+6,I)=AUX(N+5,I)                                             DHCG3250
      AUX(N+5,I)=AUX(N+3,I)                                             DHCG3260
      AUX(N+4,I)=AUX(N+1,I)                                             DHCG3270
      DELT=AUX(N+6,I)+AUX(N+5,I)                                        DHCG3280
      DELT=DELT+DELT+DELT                                               DHCG3290
  2210AUX(16,I)=8.962962962962963D0*(Y(I)-AUX(N-3,I))                  DHCG3300
     1-3.361111111111111D0*H*(DERY(I)+DELT+AUX(N+4,I))                  DHCG3310
      GOTO 201                                                          DHCG3320
C                                                                       DHCG3330
C                                                                       DHCG3340
C     H MUST BE HALVED                                                  DHCG3350
  222 IHLF=IHLF+1                                                       DHCG3360
      IF(IHLF-20)223,223,210                                            DHCG3370
  223 H=.5D0*H                                                          DHCG3380
      ISTEP=0                                                           DHCG3390
      DO 224 I=1,NDIM                                                   DHCG3400
     0Y(I)=.390625D-2*(8.D1*AUX(N-1,I)+135.D0*AUX(N-2,I)+4.D1*AUX(N-3,I)DHCG3410
     1+AUX(N-4,I))-.1171875D0*(AUX(N+6,I)-6.D0*AUX(N+5,I)-AUX(N+4,I))*H DHCG3420
     0AUX(N-4,I)=.390625D-2*(12.D0*AUX(N-1,I)+135.D0*AUX(N-2,I)+        DHCG3430
     1108.D0*AUX(N-3,I)+AUX(N-4,I))-.0234375D0*(AUX(N+6,I)+             DHCG3440
     218.D0*AUX(N+5,I)-9.D0*AUX(N+4,I))*H                               DHCG3450
      AUX(N-3,I)=AUX(N-2,I)                                             DHCG3460
  224 AUX(N+4,I)=AUX(N+5,I)                                             DHCG3470
      X=X-H                                                             DHCG3480
      DELT=X-(H+H)                                                      DHCG3490
      CALL FCT(DELT,Y,DERY)                                             DHCG3500
      DO 225 I=1,NDIM                                                   DHCG3510
      AUX(N-2,I)=Y(I)                                                   DHCG3520
      AUX(N+5,I)=DERY(I)                                                DHCG3530
  225 Y(I)=AUX(N-4,I)                                                   DHCG3540
      DELT=DELT-(H+H)                                                   DHCG3550
      CALL FCT(DELT,Y,DERY)                                             DHCG3560
      DO 226 I=1,NDIM                                                   DHCG3570
      DELT=AUX(N+5,I)+AUX(N+4,I)                                        DHCG3580
      DELT=DELT+DELT+DELT                                               DHCG3590
     0AUX(16,I)=8.962962962962963D0*(AUX(N-1,I)-Y(I))                   DHCG3600
     1-3.361111111111111D0*H*(AUX(N+6,I)+DELT+DERY(I))                  DHCG3610
  226 AUX(N+5,I)=DERY(I)                                                DHCG3620
      GOTO 206                                                          DHCG3630
      END                                                               DHCG3640
```

```
      SUBROUTINE DREBS(DFN,Y,T,N,JM,IND,JSTART,H,HMIN,EPS,R,S,WK,IER)    DTRB0780
C                                                                       DTRB0020
C-DREBS---------D-------LIBRARY 1---------------------------------------DTRB0030
C                                                                       DTRB0040
C   FUNCTION            - FIRST ORDER DIFFERENTIAL EQUATION SOLVER -    DTRB0050
C                           THE METHOD OF BULIRSCH - STOER FOR          DTRB0060
C                           DY/DT = F(Y,T)                              DTRB0070
C   USAGE               - CALL DREBS(DFN,Y,T,N,JM,IND,JSTART,H,HMIN,    DTRB0080
C                           EPS,R,S,WK,IER)                             DTRB0090
C   PARAMETERS   DFN    - USER SUPPLIED EXTERNAL SUBROUTINE,            DTRB0100
C                           DFN(Y,T,N,DY). DFN MUST CALCULATE DY(I) =   DTRB0110
C                           F(Y(1),Y(2),...,Y(N),T) FOR I=1,2,...,N.    DTRB0120
C                Y      - ON INPUT Y(1),Y(2),...,Y(N) ARE INITIAL VALUESDTRB0130
C                         ON OUTPUT Y(1),Y(2),...,Y(N) CONTAIN AN       DTRB0140
C                           APPROXIMATE SOLUTION TO Y AT T (AS SET ON   DTRB0150
C                           OUTPUT).                                    DTRB0160
C                T      - T IS THE INDEPENDENT VARIABLE.  ON INPUT, T   DTRB0170
C                           SHOULD CONTAIN THE INITIAL VALUE OF THE     DTRB0180
C                           INDEPENDENT VARIABLE. ON OUTPUT, T          DTRB0190
C                           CONTAINS THE UPDATED VALUE OF THE INDEPEN-  DTRB0200
C                           DENT VARIABLE.                              DTRB0210
C                N      - THE NUMBER OF EQUATIONS IN THE SYSTEM         DTRB0220
C                JM     - THE MAXIMUM ORDER OF THE RATIONAL APPROX-     DTRB0230
C                           IMATION. JM MUST BE LESS THAN 7.            DTRB0240
C                IND    - CONVERGENCE TYPE INDICATOR (INPUT)            DTRB0250
C                           IND = 1 SPECIFIES THE STANDARD ERROR TEST   DTRB0260
C                           IND = 2 SPECIFIES THE RELATIVE ERROR TEST   DTRB0270
C                           IND = 3 SPECIFIES THE ABSOLUTE ERROR TEST   DTRB0280
C                JSTART - AN INPUT INDICATOR WITH THE FOLLOWING MEANINGSDTRB0290
C                           0 - PERFORM A STEP. THE FIRST STEP          DTRB0300
C                              MUST BE DONE WITH THIS VALUE OF JSTART SO DTRB0310
C                              THAT THE SUBROUTINE CAN INITIALIZE ITSELF.DTRB0320
C                          -1 - REPEAT THE LAST STEP WITH A NEW VALUE   DTRB0330
C                              OF H OR JM. THE INITIAL VALUES OF Y, S,  DTRB0340
C                              AND T ARE SET TO THE INITIAL VALUES OF Y, DTRB0350
C                              S, AND T FROM THE MOST RECENT CALL TO    DTRB0360
C                              DREBS WITH JSTART = 0.                   DTRB0370
C                H      - ON INPUT, H IS AN INITIAL GUESS FOR THE STEP  DTRB0380
C                           SIZE.                                       DTRB0390
C                         ON OUTPUT, H CONTAINS A SUGGESTED STEP SIZE   DTRB0400
C                           FOR THE NEXT STEP. THE SUGGESTED VALUE MAY  DTRB0410
C                           BE LARGER OR SMALLER THAN THE ORIGINAL      DTRB0420
C                           STEP SIZE.                                  DTRB0430
C                HMIN   - HMIN IS THE SMALLEST PERMISSIBLE STEP SIZE.   DTRB0440
C                           DREBS WILL DECREASE THE STEP SIZE           DTRB0450
```

```
C                   UNTIL CONVERGENCE CAN BE OBTAINED.                 DTRB0460
C            EPS    - ERROR TOLERANCE                                  DTRB0470
C            R      - ON OUTPUT, THE N-VECTOR R CONTAINS THE           DTRB0480
C                       ABSOLUTE ERRORS IN EACH COMPONENT FOR          DTRB0490
C                       THE CURRENT STEP.                              DTRB0500
C            S      - AN N-VECTOR CONTAINING EITHER (1) THE LARGEST    DTRB0510
C                       VALUE OF EACH Y COMPUTED SINCE THE START OF    DTRB0520
C                       THE INTEGRATION FOR THE STANDARD ERROR TEST,   DTRB0530
C                       (2) THE LARGEST VALUE OF EACH Y COMPUTED       DTRB0540
C                       DURING THE CURRENT STEP BEING TAKEN FOR        DTRB0550
C                       RELATIVE ERROR TEST, OR (3) THE VALUE 1.0      DTRB0560
C                       FOR THE ABSOLUTE ERROR TEST.                   DTRB0570
C                       S MUST BE INITIALIZED IN ACCORDANCE WITH       DTRB0580
C                       THE ERROR TEST SELECTED BEFORE THE FIRST       DTRB0590
C                       CALL TO DREBS. IF IND = 1 OR 2, S(I) SHOULD    DTRB0600
C                       EQUAL Y(I) ON THE FIRST CALL. IF IND = 3,      DTRB0610
C                       S(I) SHOULD EQUAL 1.                           DTRB0620
C            WK     - WORKING STORAGE OF DIMENSION 29*N                DTRB0630
C            IER    - ERROR INDICATOR                                  DTRB0640
C                     TERMINAL ERROR = 128 + N                         DTRB0650
C                       N = 1 INDICATES CONVERGENCE COULD NOT BE       DTRB0660
C                         OBTAINED WITH CURRENT VALUES OF H AND        DTRB0670
C                         HMIN, Y,T, AND H HAVE BEEN UPDATED.          DTRB0680
C                     WARNING ERROR (WITH FIX) = 64 + N                DTRB0690
C                       N = 2 INDICATES JM IS LESS THAN 1 OR GREATERDTRB0700
C                         THAN 6. SET JM = 6.                          DTRB0710
C   PRECISION           - SINGLE/DOUBLE                                DTRB0720
C   REQD. IMSL ROUTINES - UERTST                                       DTRB0730
C   LANGUAGE            - FORTRAN                                      DTRB0740
C----------------------------------------------------------------------DTRB0750
C   LATEST REVISION     - APRIL 5, 1977                                DTRB0760
C                                                                      DTRB0770
C                                                                      DTRB0790
      DIMENSION          Y(N),S(N),R(N),D(7),WK(116)                   DTRB0 00
      DOUBLE PRECISION   Y,T,H,HMIN,EPS,R,S,WK,D,ZOTUP,A,G,            DTRB0810
     1                   B,XU,U,C,TA,B1,V,UST,TOLD,HALF                DTRB08
     2                   ,DABS,DFLOAT
      DATA               ZOTUP/Z4D40000000000000/,HALF/.5D0/           DTRB0830
      LOGICAL            KONVF,KONV,BO,BH                              DTRB0850
      IER=0                                                            DTRB0860
      N2=N+N                                                           DTRB0870
      N3=N2+N                                                          DTRB0880
      N4=N3+N                                                          DTRB0890
      N5=N4+N                                                          DTRB0900
      N6=N5+N5+N2                                                      DTRB0910
```

```
      N7=N6+N5+N3                                                       DTRB0920
      N8=N7+N5+N3                                                       DTRB0930
      N2P1=N2+1                                                         DTRB0940
      N3P1=N3+1                                                         DTRB0950
      N8P1=N8+1                                                         DTRB0960
      IF(JM.GT.0.AND.JM.LE.6) GO TO 5                                   DTRB0970
      IER = 66                                                          DTRB0980
      JM=6                                                              DTRB0990
C                                  FOR AN EXTRAPOLATION OF ORDER JM,    DTRB1000
C                                  JM+1 APPROXIMATIONS ARE REQUIRED.    DTRB1010
C                                  THREE MORE ARE ALLOWED IN ATTEMPTING DTRB1020
C                                  TO ACHIEVE CONVERGENCE.              DTRB1030
    5 JMAX = JM+4                                                       DTRB1040
      IF(JSTART.NE.0) GO TO 135                                         DTRB1050
C                                  INITIALIZATION                       DTRB1060
C                                  SAVE THE INITIAL VALUES FOR THE DEP- DTRB1070
C                                  ENDENT VARIABLES AND THE ERROR TEST  DTRB1080
C                                  VECTOR FOR THE STEP                  DTRB1090
      TOLD = T                                                          DTRB1100
      IJK4=N4                                                           DTRB1110
      DO 10 I = 1,N                                                     DTRB1120
         WK(I) = Y(I)                                                   DTRB1130
         IJK4=IJK4+1                                                    DTRB1140
         WK(IJK4) = S(I)                                                DTRB1150
   10 CONTINUE                                                          DTRB1160
C                                  USE THE FUNCTION ROUTINE TO OBTAIN   DTRB1170
C                                  THE INITIAL SLOPES                   DTRB1180
C                                  DZ = DY/DX                           DTRB1190
      CALL DFN(Y,T,N,WK(N3P1))                                          DTRB1200
C                                  THE LOGICAL VARIABLE,BH, DETERMINES  DTRB1210
C                                  WHETHER THE STEPSIZE HAS BEEN        DTRB1220
C                                  HALVED, INITIALLY FALSE.             DTRB1230
C                                  LATER BH IS FALSE IF THE STEPSIZE IS DTRB1240
C                                  CUT BY A FACTOR NOT 2                DTRB1250
   15 BH = .FALSE.                                                      DTRB1260
C                                  PRESET THE CONVERGENCE SUCCESS FLAG  DTRB1270
C                                     TRUE                              DTRB1280
      KONVF = .TRUE.                                                    DTRB1290
C                                  ADVANCE THE INDEPENDENT VARIABLE BY  DTRB1300
C                                  THE STEPSIZE, H                      DTRB1310
   20 A = H+T                                                           DTRB1320
C                                  SET THE SWITCH BO FOR THE FIRST SET  DTRB1330
C                                  OF COEFFICIENTS, D                   DTRB1340
      BO = .FALSE.                                                      DTRB1350
C                                  INITIALIZE THE H SEQUENCE...         DTRB1360
```

```
C                                         H/M,H/JR,H/JS                        DTRB1370
      M = 1                                                                    DTRB1380
      JR = 2                                                                   DTRB1390
      JS = 3                                                                   DTRB1400
C                                         JJ IS THE INDEX FOR THE ARRAY OF     DTRB1410
C                                         VALUES SAVED IN CASE THE INTERVAL    DTRB1420
C                                         MUST BE HALVED                       DTRB1430
      JJ = 0                                                                   DTRB1440
C                                         INTEGRATION STEP                     DTRB1450
C                                         MIDPOINT + EXTRAPOLATION             DTRB1460
      IJ6=N6-N                                                                 DTRB1470
      IJ7=N7-N                                                                 DTRB1480
      I6=IJ6                                                                   DTRB1490
      I7=IJ7                                                                   DTRB1500
      DO 125 J = 1,JMAX                                                        DTRB1510
C                                         SET THE VALUES OF THE EXTRAPOLATION  DTRB1520
C                                         COEFFICIENTS TO THEIR CORRECT VALUES DTRB1530
C                                         FOR THIS EXTRAPOLATION STEP          DTRB1540
         IJ6=IJ6+N                                                             DTRB1550
         IJ7=IJ7+N                                                             DTRB1560
         IF (.NOT. BO) GO TO 25                                                DTRB1570
         D(2)=1.7777777777778D0                                                DTRB1580
         D(4)=7.1111111111111D0                                                DTRB1590
         D(6)=28.444444444444D0                                                DTRB1600
         GO TO 30                                                              DTRB1640
   25    D(2)=2.25D0                                                           DTRB1650
         D(4)=9.D0                                                             DTRB1660
         D(6)=36.D0                                                            DTRB1670
C                                         IF THE ORDER OF THE EXTRAPOLATION    DTRB1710
C                                         STEP BEING COMPUTED IS LESS THAN JM/2DTRB1720
C                                         SET KONV FALSE                       DTRB1730
   30    KONV = .TRUE.                                                         DTRB1740
         IF (J .LE. (JM/2)) KONV = .FALSE.                                     DTRB1750
         IF (J .LE. (JM+1)) GO TO 35                                           DTRB1760
C                                         RESTRICT THE ORDER OF THE EXTRAPOL-  DTRB1770
C                                         ATION TO JM                          DTRB1780
C                                         ADJUST THE EXTRAPOLATION COEFFICIENT DTRB1790
         L = JM+1                                                              DTRB1800
         D(L) = 4.D0*D(L-2)                                                    DTRB1810
C                                         DISCOURAGE THE STEP-INCREASING FACTORDTRB1830
C                                         FC, BY A FACTOR OF SQRT(2) SINCE     DTRB1840
C                                         CONVERGENCE WAS NOT OBTAINED         DTRB1850
C                                         IN JM EXTRAPOLATIONS                 DTRB1860
         FC = .7071068*FC                                                      DTRB1870
         GO TO 40                                                              DTRB1880
```

```
      C                                        THE NUMBER, J, OF EXTRAPOLATIONS HAS DTRB1890
      C                                        NOT EXCEEDED JM                      DTRB1900
      C                                        FIND D(J) = ((H DIVIDED BY H/M)**2   DTRB1910
      C                                        ADJUST THE FACTOR,FC, USED TO ADJUST DTRB1920
      C                                        THE STEPSIZE FOR THE NEXT STEP TO BE DTRB1930
      C                                        TAKEN                                DTRB1940
   35     L = J                                                                     DTRB1950
          D(L) = DFLOAT(M*M)                                                        DTRB1960
          FC = 1.0+FLOAT(JM+1-J)*.1666667                                           DTRB1980
      C                                        MODIFIED MIDPOINT RULE USED TO FIND  DTRB1990
      C                                        FIRST VALUE FOR THIS EXTRAPOLATION   DTRB2000
      C                                        STEP                                 DTRB2010
   40     M = M+M                                                                   DTRB2020
          G = H/FLOAT(M)                                                            DTRB2030
          B = G+G                                                                   DTRB2040
      C                                        IF THE STEPSIZE HAS NOT BEEN HALVED  DTRB2050
      C                                        OR IF THE ORDER OF THE EXTRAPOLATION DTRB2060
      C                                        STEP EXCEEDS THAT FOR WHICH PREVIOUS-DTRB2070
      C                                        LY COMPUTED VALUES WERE SAVED, THEY  DTRB2080
      C                                        MUST BE COMPUTED                     DTRB2090
          IF ((.NOT. BH) .OR. (J .GE. (JMAX-1))) GO TO 50                           DTRB2100
      C                                        OTHERWISE THE VALUES HAVE BEEN SAVED DTRB2110
      C                                        AND CAN BE RESTORED                  DTRB2120
          IJK1=N                                                                    DTRB2130
          IJK2=N2                                                                   DTRB2140
          IJK6=IJ6                                                                  DTRB2150
          IJK7=IJ7                                                                  DTRB2160
          DO 45 I = 1,N                                                             DTRB2170
             IJK2=IJK2+1                                                            DTRB2180
             IJK7=IJK7+1                                                            DTRB2190
             WK(IJK2) = WK(IJK7)                                                    DTRB2200
             IJK1=IJK1+1                                                            DTRB2210
             IJK6=IJK6+1                                                            DTRB2220
             WK(IJK1) = WK(IJK6)                                                    DTRB2230
   45     CONTINUE                                                                  DTRB2240
          GO TO 75                                                                  DTRB2250
      C                                        COMPUTE STARTING VALUES FOR THE MODI-DTRB2260
      C                                        FIED MIDPOINT RULE                   DTRB2270
   50     IJK1=N                                                                    DTRB2280
          IJK2=N2                                                                   DTRB2290
          IJK3=N3                                                                   DTRB2300
          DO 55 I = 1,N                                                             DTRB2310
             IJK1=IJK1+1                                                            DTRB2320
             WK(IJK1) = WK(I)                                                       DTRB2330
             IJK2=IJK2+1                                                            DTRB2340
```

```
            IJK3=IJK3+1                                                 DTRB2350
            WK(IJK2) = WK(I)+G*WK(IJK3)                                 DTRB2360
   55    CONTINUE                                                       DTRB2370
         KH = M/2                                                       DTRB2380
         XU=1                                                           DTRB2390
C                                      THE MEMBER OF THE H SEQUENCE     DTRB2400
C                                      BEING USED BY THE MIDPOINT INTEGRA-  DTRB2410
C                                      TION RULE IS H/M. COMPUTE THE END    DTRB2420
C                                      OF THE STEP  FOR EACH DEPENDENT VARI-DTRB2430
C                                      ABLE                             DTRB2440
         DO 70 K = 2,M                                                  DTRB2450
            XU = XU + G                                                 DTRB2460
            CALL DFN(WK(N2P1),XU,N,WK(N8P1))                            DTRB2470
            IJK1=N                                                      DTRB2480
            IJK2=N2                                                     DTRB2490
            IJK8=N8                                                     DTRB2500
            DO   60 I = 1,N                                             DTRB2510
               IJK1=IJK1+1                                              DTRB2520
               IJK8=IJK8+1                                              DTRB2530
               U = WK(IJK1) + B*WK(IJK8)                                DTRB2540
               IJK2=IJK2+1                                              DTRB2550
               WK(IJK1) = WK(IJK2)                                      DTRB2560
               WK(IJK2) = U                                             DTRB2570
   60       CONTINUE                                                    DTRB2580
C                                      IN CASE THE INTERVAL MUST BE HALVED  DTRB2590
C                                      NEXT TIME, SAVE THE VALUES AT HALFWAYDTRB2600
C                                      ALONG (KH=M/2) THE STEP UNLESS K=3   DTRB2610
            IF ((K .NE. KH) .OR. (K .EQ. 3)) GO TO 70                   DTRB2620
            JJ = 1+JJ                                                   DTRB2630
            I6=I6+N                                                     DTRB2640
            I7=I7+N                                                     DTRB2650
            IJK1=N                                                      DTRB2660
            IJK2=N2                                                     DTRB2670
            IJK6=I6                                                     DTRB2680
            IJK7=I7                                                     DTRB2690
            DO 65 I = 1,N                                               DTRB2700
               IJK2=IJK2+1                                              DTRB2710
               IJK7=IJK7+1                                              DTRB2720
               WK(IJK7) = WK(IJK2)                                      DTRB2730
               IJK6=IJK6+1                                              DTRB2740
               IJK1=IJK1+1                                              DTRB2750
               WK(IJK6) = WK(IJK1)                                      DTRB2760
   65       CONTINUE                                                    DTRB2770
   70    CONTINUE                                                       DTRB2780
   75    CALL DFN(WK(N2P1),A,N,WK(N8P1))                                DTRB2790
```

```
      IJK1=N                                                            DIRB2800
      IJK2=N2                                                           DIRB2810
      IJK5=N5                                                           DIRB2820
      IK5=N5                                                            DIRB2830
      IJK8=N8                                                           DIRB2840
      DO 115 I = 1,N                                                    DIRB2850
C                                   V IS USED TO SAVE THE VALUE OBTAINED DIRB2860
C                                   BY THE MIDPOINT RULE USING THE PREVI-DIRB2870
C                                   OUS MEMBER OF THE H SEQUENCE        DIRB2880
C                                   (THE FIRST TIME THROUGH THIS VALUE ISDIRB2890
C                                   MEANINGLESS, BUT IT IS NOT USED SINCEDIRB2900
C                                   L IS LESS THAN 2)                   DIRB2910
         IJK1=IJK1+1                                                    DIRB2920
         IJK2=IJK2+1                                                    DIRB2930
         IJK5=IJK5+1                                                    DIRB2940
         IK5=IK5+1                                                      DIRB2950
         IF (L .GE. 2) V = WK(IJK5)                                     DIRB2960
C                                   COMPUTE THE FINAL VALUE OBTAINED FOR DIRB2970
C                                   THIS MEMBER OF THE H SEQUENCE BY THE DIRB2980
C                                   MODIFIED MIDPOINT RULE              DIRB2990
         IJK8=IJK8+1                                                    DIRB3000
         WK(IJK5) = (WK(IJK2) + WK(IJK1) + G * WK(IJK8))*HALF           DIRB3010
         C = WK(IJK5)                                                   DIRB3020
         TA = C                                                         DIRB3030
C                                   AT LEAST TWO VALUES ARE NEEDED TO   DIRB3040
C                                   START EXTRAPOLATION                 DIRB3050
         IF (L .LT. 2) GO TO 90                                         DIRB3060
C                                   IF THE VALUE JUST COMPUTED BY THE   DIRB3070
C                                   MIDPOINT RULE SHOWS A LARGE JUMP FROMDIRB3080
C                                   THE PREVIOUS, HALVE THE INTERVAL    DIRB3090
         IF((DABS(V)*2DTUP .LT. DABS(C)) .AND. (H .NE. HMIN) .AND.      DIRB3100
     1      (J.GT.JM/2+1)) GO TO 130                                    DIRB3120
C                                   PERFORM THE L STEPS FOR THE CURRENT DIRB3130
C                                   LTH ORDER EXTRAPOLATION STEP. IF THE DIRB3140
C                                   DENOMINATOR OF THE RATIONAL FUNCTION DIRB3150
C                                   GOES TO ZERO AT ANY STEP, SET DT AT DIRB3160
C                                   THAT STEP TO ITS VALUE JUST BEFORE  DIRB3170
         IP5=IK5                                                        DIRB3180
         DO 85 K = 2,L                                                  DIRB3190
            IP5=IP5+N                                                   DIRB3200
            B1 = D(K) * V                                               DIRB3210
            B = B1-C                                                    DIRB3220
            U = V                                                       DIRB3230
            IF (B .EQ. 0.) GO TO 80                                     DIRB3240
            B = (C-V)/B                                                 DIRB3250
```

```
              U = C*B                                          DTRB3260
              C = B1*B                                         DTRB3270
   80         IF(K.LT.L) V=WK(IPS)                             DTRB3280
              WK(IPS) = U                                      DTRB3290
              TA = U + TA                                      DTRB3300
   85      CONTINUE                                            DTRB3310
C                              USE THE ERROR ROUTINE FOR EACH  DTRB3320
C                              DEPENDENT VARIABLE TO CHECK WHETHER DTRB3330
C                              CONVERGENCE HAS BEEN ACHIEVED   DTRB3340
   90      GO TO (95,100,105),IND                              DTRB3350
   95      UST=DABS(TA)                                        DTRB3360
           IF(UST.GT.S(I)) S(I)=UST                            DTRB3380
           GO TO 110                                           DTRB3390
  100      S(I)=DABS(Y(I))                                     DTRB3400
           GO TO 110                                           DTRB3420
  105      S(I)=1.                                             DTRB3430
  110      R(I)=DABS(Y(I)-TA)                                  DTRB3440
           Y(I)=TA                                             DTRB3460
           IF(S(I).LT.EPS) S(I)=EPS                            DTRB3470
           IF(R(I).GT.EPS*S(I)) KONV=.FALSE.                   DTRB3480
  115    CONTINUE                                              DTRB3490
         IF(KONV) GO TO 155                                    DTRB3500
C                              RESET THE EXTRAPOLATION COEFFICIENTS DTRB3510
         D(3) = 4.                                             DTRB3520
         D(5) = 16.                                            DTRB3530
C                              FLIP THE BO SWITCH FOR THE NEXT SET DTRB3540
C                              OF COEFFICIENTS                 DTRB3550
         BO = (.NOT.BO)                                        DTRB3560
C                              RESET S                         DTRB3570
         IJK4=N4                                               DTRB3580
         DO 120 I=1,N                                          DTRB3590
            IJK4=IJK4+1                                        DTRB3600
            S(I)=WK(IJK4)                                      DTRB3610
  120    CONTINUE                                              DTRB3620
C                              TAKE THE NEXT MEMBER            DTRB3630
C                              OF THE H SEQUENCE               DTRB3640
         M = JR                                                DTRB3650
         JR = JS                                               DTRB3660
         JS = M+M                                              DTRB3670
C                              AND GO BACK FOR THE             DTRB3680
C                              NEXT EXTRAPOLATION              DTRB3690
  125 CONTINUE                                                 DTRB3700
C                              IF, AFTER ALL THE EXTRAPOLATIONS DTRB3710
C                              ALLOWED, CONVERGENCE HAS NOT BEEN DTRB3720
C                              ACHIEVED, ATTEMPT TO HALVE H SO THAT DTRB3730
```

```
C                                          THE SAVED VALUES CAN BE USED (SET BH DTRB3740
C                                          TRUE FOR THIS PURPOSE)               DTRB3750
C                                          IF HALVING H MAKES IT LESS THAN HMIN,DTRB3760
C                                          SET H = HMIN                         DTRB3770
C                                          IN THIS CASE THE SAVED VALUES CANNOT DTRB3780
C                                          BE USED                              DTRB3790
C                                          IF H HAD ALREADY BEEN AT HMIN,       DTRB3800
C                                          CONVERGENCE CANNOT BE ACHIEVED FOR   DTRB3810
C                                          THIS HMIN AND THIS EPS CRITERION.    DTRB3820
C                                          SET KONV FALSE                       DTRB3830
      BH = (.NOT. BH)                                                           DTRB3840
  130 IF (DABS(H) .LE. DABS(HMIN)) GO TO 150                                    DTRB3850
      H = H * HALF                                                              DTRB3870
      IF (DABS(H) .GE. DABS(HMIN)) GO TO 20                                     DTRB3880
      H = DSIGN(HMIN,H)                                                         DTRB3900
      GO TO 15                                                                  DTRB3920
  135 DO 140 I = 1,N                                                            DTRB3930
         Y(I) = WK(I)                                                           DTRB3940
  140 CONTINUE                                                                  DTRB3950
      IJK4=N4                                                                   DTRB3960
      DO 145 I=1,N                                                              DTRB3970
         IJK4=IJK4+1                                                            DTRB3980
         S(I)=WK(IJK4)                                                          DTRB3990
  145 CONTINUE                                                                  DTRB4000
      T = TOLD                                                                  DTRB4010
      GO TO 15                                                                  DTRB4020
  150 KONVF = .FALSE.                                                           DTRB4030
C                                          WHETHER OR NOT CONVERGENCE HAS BEEN  DTRB4040
C                                          ACHIEVED SET A NEW SUGGESTED STEPSIZEDTRB4050
C                                          FOR THE NEXT STEP                    DTRB4060
C                                          ASSIGN THE END OF STEP VALUE TO THE  DTRB4070
C                                          INDEPENDENT VARIABLE                 DTRB4080
  155 H = FC*H                                                                  DTRB4090
      T=A                                                                       DTRB4100
      IF(KONVF) GO TO 160                                                       DTRB4110
      IER=129                                                                   DTRB4120
  160 IF(IER.EQ.0) GO TO 9005                                                   DTRB4130
 9000 CONTINUE                                                                  DTRB4140
      CALL JERIST (IER,6HDREBS )                                                DTRB4150
 9005 RETURN                                                                    DTRB4160
      END                                                                       DTRB4170
```

```
      SUBROUTINE UERTST(IER,NAME)
C
      DIMENSION          ITYP(5,4),IBIT(4)
      INTEGER*2          NAME(3)
      INTEGER            WARN,WARF,TERM,PRINTR
      EQUIVALENCE        (IBIT(1),WARN),(IBIT(2),WARF),(IBIT(3),TERM)
      DATA      ITYP     /'WARN','ING ','    ','    ','    ',
     *                     'WARN','ING(','WITH',' FIX',')   ',
     *                     'TERM','INAL','    ','    ','    ',
     *                     'NON-','DEFI','NED ','    ','    '/,
     *          IBIT     / 32,64,128,0/
      DATA      PRINTR   / 6/
      IER2=IER
      IF (IER2 .GE. WARN) GO TO 5
C                                       NON-DEFINED
      IER1=4
      GO TO 20
    5 IF (IER2 .LT. TERM) GO TO 10
C                                       TERMINAL
      IER1=3
      GO TO 20
   10 IF (IER2 .LT. WARF) GO TO 15
C                                       WARNING(WITH FIX)
      IER1=2
      GO TO 20
C                                       WARNING
   15 IER1=1
C                                       EXTRACT 'N'
   20 IER2=IER2-IBIT(IER1)
C                                       PRINT ERROR MESSAGE
      WRITE (PRINTR,25) (ITYP(I,IER1),I=1,5),NAME,IER2,IER
   25 FORMAT(' *** I M S L(UERTST) ***  ',5A4,4X,3A2,4X,I2,
     *   ' (IER = ',I3,')')
      RETURN
      END
```

```
      SUBROUTINE DVOGER(DFUN,Y,T,N,MTH,MAXDER,JSTART,H,HMIN,HMAX,EPS,    DVOG1300
C    *                YMAX,ERROR,WK,IER)                                  DVOG0020
C-DVOGER--------D--------LIBRARY 1----------------------------------------DVOG0030
C                                                                         DVOG0040
C   FUNCTION            - FIRST ORDER DIFFERENTIAL EQUATION SOLVER-      DVOG0050
C                           GEAR'S METHOD FOR DX/DT=F(X,T)                DVOG0060
C   USAGE               - CALL DVOGER(DFUN,Y,T,N,MTH,MAXDER,JSTART,H,    DVOG0070
C                           HMIN,HMAX,EPS,YMAX,ERROR,WK,IER)              DVOG0080
C   PARAMETERS   DFUN   - USER SUPPLIED EXTERNAL SUBROUTINE, DFUN(YP,TP,DVOG0090
C                           M,DY,PW,IND), WHERE                           DVOG0100
C                             YP CONTAINS THE PRESENT (I.E. AT TP)        DVOG0110
C                               SOLUTION VECTOR AS X(1)=YP(1,1),          DVOG0120
C                               X(2)=YP(1,2),...,X(N)=YP(1,N),            DVOG0130
C                             TP IS THE PRESENT TIME,AND                  DVOG0140
C                             M IS THE ORDER OF THE JACOBIAN.             DVOG0150
C                           IF IND=0, DFUN MUST COMPUTE THE N-VECTOR      DVOG0160
C                             F(YP,TP) AND STORE THE VALUES IN DY.        DVOG0170
C                           IF IND=1, DFUN MUST COMPUTE THE JACOBIAN OF  DVOG0180
C                             F EVALUATED AT (YP,TP) AND STORE THE        DVOG0190
C                             RESULT IN THE M BY M MATRIX PW.             DVOG0200
C                           THE JACOBIAN IS COMPUTED ONLY FOR CALLS WITHDVOG0210
C                             MTH = 1.                                    DVOG0220
C                         YP IS AN 8 BY N ARRAY. THE SOLUTION COMPONENTSDVOG0230
C                           ARE Y(1,1),Y(1,2),...,Y(1,N).                 DVOG0240
C                Y      - Y IS A TWO DIMENSIONAL ARRAY (8 BY N)           DVOG0250
C                           CONTAINING THE DEPENDENT VARIABLES AND THEIRDVOG0260
C                           SCALED DERIVATIVES. THE SOLUTION COMPONENTS  DVOG0270
C                           ARE X(1) = Y(1,1), X(2) = Y(1,2),...,         DVOG0280
C                           X(N) = Y(1,N) AND Y(J+1,I) CONTAINS THE J-THDVOG0290
C                           DERIVATIVE OF X(I) SCALED BY H**J/FACTORIAL  DVOG0300
C                           (J). HERE H IS THE CURRENT STEP SIZE.        DVOG0310
C                         ONLY Y(1,I), I=1,2,...,N NEED BE PROVIDED BY   DVOG0320
C                           THE CALLING PROGRAM ON THE FIRST CALL TO     DVOG0330
C                           DVOGER, I.E. WITH JSTART = 0.                DVOG0340
C                T      - T IS THE INDEPENDENT VARIABLE.  ON INPUT, T    DVOG0350
C                           SHOULD CONTAIN THE INITIAL VALUE OF THE      DVOG0360
C                           INDEPENDENT VARIABLE.  ON OUTPUT, T          DVOG0370
C                           CONTAINS THE UPDATED VALUE OF THE INDEPEN-   DVOG0380
C                           DENT VARIABLE.                               DVOG0390
C                N      - N IS THE NUMBER OF FIRST ORDER DIFFERENTIAL    DVOG0400
C                           EQUATIONS.                                   DVOG0410
C                MTH    - MTH IS THE METHOD INDICATOR. THE USER MAY      DVOG0420
C                           SELECT ONE OF THE FOLLOWING.                 DVOG0430
C                             MTH=0 INDICATES A PREDICTOR-CORRECTOR      DVOG0440
C                               (ADAMS) METHOD                           DVOG0450
```

```
C                  MTH=1 INDICATES A VARIABLE-ORDER METHOD    DVOG0460
C                     SUITABLE FOR SYSTEMS OF STIFF DIFFER-   DVOG0470
C                     ENTIAL EQUATIONS. HERE THE USER MUST    DVOG0480
C                     PROVIDE THE PARTIAL DERIVATIVES OF THE  DVOG0490
C                     DIFFERENTIAL EQUATIONS WITH RESPECT TO  DVOG0500
C                     THE X'S, THAT IS, THE JACOBIAN. THIS IS DVOG0510
C                     DONE IN THE CALL TO DFUN(YP,TP,M,DY,PW, DVOG0520
C                     IND) WITH IND=1. ON OUTPUT, PW(I,J)     DVOG0530
C                     (I,J=1,2,...,M) MUST BE SET TO THE      DVOG0540
C                     PARTIAL DERIVATIVE OF THE I-TH EQUATION DVOG0550
C                     WITH RESPECT TO X(J).                   DVOG0560
C                  MTH=2 INDICATES A VARIABLE-ORDER METHOD,   DVOG0570
C                     BUT DVOGER COMPUTES THE PARTIAL DERVA-  DVOG0580
C                     TIVES BY NUMERICAL DIFFERENCING. HENCE, DVOG0590
C                     DFUN IS CALLED WITH IND=0 ONLY.         DVOG0600
C        MAXDER - MAXDER MUST BE SET BY THE USER TO THE MAXIMUM DVOG0610
C                  ORDER TO BE USED IN THE APPROXIMATION.     DVOG0620
C                  IT MUST BE LESS THAN 8 FOR THE ADAMS METHOD DVOG0630
C                  AND LESS THEN 7 FOR THE STIFF METHODS.     DVOG0640
C        JSTART - ON INPUT, JSTART HAS THE FOLLOWING MEANINGS...DVOG0650
C                  -1 - REPEAT THE LAST STEP WITH A NEW VALUE DVOG0660
C                     FOR H.                                  DVOG0670
C                   0 - INITIALIZE THE INTEGRATION, THE FIRST DVOG0680
C                     CALL TO DVOGER MUST BE DONE WITH THIS   DVOG0690
C                     VALUE OF JSTART.                        DVOG0700
C                  +1 - TAKE A NEW STEP CONTINUING FROM THE   DVOG0710
C                     LAST.                                   DVOG0720
C                 ON OUTPUT, JSTART IS SET TO 10, THE CURRENT DVOG0730
C                  ORDER OF THE METHOD. JSTART IS ALSO THE    DVOG0740
C                  ORDER OF THE MAXIMUM DERIVATIVE AVAILABLE INDVOG0750
C                  Y.                                         DVOG0760
C        H      - ON INPUT, H CONTAINS THE STEP SIZE TO BE    DVOG0770
C                  ATTEMPTED ON THE NEXT STEP, IF THIS STEP   DVOG0780
C                  SIZE DOES NOT CAUSE A LARGER ERROR THAN    DVOG0790
C                  REQUESTED, IT WILL BE USED. OTHERWISE, SEE DVOG0800
C                  PARAMETER IER. THE USER IS ADVISED TO USE  DVOG0810
C                  A FAIRLY SMALL STEP IN THE FIRST CALL TO   DVOG0820
C                  DVOGER.                                    DVOG0830
C                 ON OUTPUT, H CONTAINS A SUGGESTED STEP SIZE DVOG0840
C                  FOR THE NEXT STEP IN ORDER TO ACHIEVE AN   DVOG0850
C                  ECONOMICAL INTEGRATION.                    DVOG0860
C        HMIN   - HMIN MUST BE SET TO THE SMALLEST STEP SIZE  DVOG0870
C                  ALLOWABLE IN THIS INTEGRATION. HMIN SHOULD DVOG0880
C                  BE MUCH SMALLER THAN THE EXPECTED AVERAGE  DVOG0890
C                  STEP SIZE  FOR.THE FIRST CALL SINCE A FIRST DVOG0900
```

```
C                                 ORDER METHOD IS USED INITIALLY.            DVOG0910
C                     HMAX      - HMAX MUST BE SET TO THE LARGEST STEP SIZE  DVOG0920
C                                 ALLOWABLE IN THIS INTEGRATION.             DVOG0930
C                     EPS       - EPS IS USED TO SPECIFY THE MAXIMUM ERROR   DVOG0940
C                                 CRITERION. THE STEP SIZE AND/OR THE ORDER ISDVOG0950
C                                 ADJUSTED SO THAT THE SINGLE STEP ERROR ESTI-DVOG0960
C                                 MATES DIVIDED BY YMAX(I) ARE LESS THAN     DVOG0970
C                                 EPS IN THE EUCLIDEAN NORM.                 DVOG0980
C                     YMAX      - YMAX IS AN N-VECTOR WHICH CONTAINS THE MAXIMUMDVOG0990
C                                 ABSOLUTE VALUE OF EACH COMPONENT OF X CAL- DVOG1000
C                                 CULATED SO FAR. THE COMPONENTS OF YMAX     DVOG1010
C                                 SHOULD NORMALLY BE SET TO 1. BEFORE THE    DVOG1020
C                                 FIRST CALL TO DVOGER.                      DVOG1030
C                     ERROR     - ERROR IS AN N-VECTOR WHICH CONTAINS THE ESTI- DVOG1040
C                                 MATED ONE STEP ERROR IN EACH COMPONENT.    DVOG1050
C                     WK        - WORK AREA OF DIMENSION                     DVOG1060
C                                 17*N       IF MTH = 0                      DVOG1070
C                                 N*(N+17) OTHERWISE                         DVOG1080
C                     IER       - ERROR PARAMETER                            DVOG1090
C                               WARNING ERROR = 32 + N                       DVOG1100
C                                 N = 1 INDICATES THE STEP WAS TAKEN WITH    DVOG1110
C                                   H=HMIN, BUT THE REQUESTED ERROR WAS NOT  DVOG1120
C                                   ACHIEVED                                 DVOG1130
C                                 N = 2 INDICATES CORRECTOR CONVERGENCE COULD DVOG1140
C                                   NOT BE ACHIEVED FOR H GREATER THAN HMIN  DVOG1150
C                                 N = 3 INDICATES THE REQUESTED ERROR IS     DVOG1160
C                                   SMALLER THAN CAN BE HANDLED FOR THIS     DVOG1170
C                                   PROBLEM                                  DVOG1180
C                               WARNING ERROR (WITH FIX ) = 64 + N           DVOG1190
C                                 N = 4 INDICATES THE MAXIMUM ORDER SPECIFIED DVOG1200
C                                   WAS FOUND TO BE TOO LARGE. THE MAXIMUM   DVOG1210
C                                   ORDER WAS SET TO 7 FOR MTH = 0 AND       DVOG1220
C                                   TO 6 OTHERWISE.                          DVOG1230
C   PRECISION                   SINGLE/DOUBLE                                DVOG1240
C   REQD. IMSL ROUTINES       - LUDATF,LUELMF,UERTST                         DVOG1250
C   LANGUAGE                  - FORTRAN                                      DVOG1260
C-----------------------------------------------------------------------------DVOG1270
C   LATEST REVISION           - MARCH 18, 1975                               DVOG1280
C                                                                            DVOG1290
     *                          YMAX,ERROR,WK,IER)                           DVOG1310
C                                                                            DVOG1320
      DIMENSION                 Y(8,N),YMAX(N),ERROR(N),WK(84,4),            DVOG1330
     1                          C(8),COEF(7,2,3)                             DVOG1340
      DOUBLE PRECISION          C,D,E,H,R,T,Y,R1,BND,EPS,EUP,EDWN,ENQ1,D1,D2, DVOG1350
     1                          ENQ2,ENQ3,HMAX,HMIN,HNEW,HOLD,TOLD,YMAX,     DVOG1360
```

```
     2                    ERROR,RACUM,WK,XK,ZERO,HALF,ONE,ONEP              DVOG1370
     3                   ,DABS,DMAX1,DMIN1
      DATA                C(2),ZERO,HALF,ONE,ONEP,NDIG/-1.D0,0.D0,.5D0,     DVOG1380
     *                    1.D0,1.00000000000001D0,0/                        DVOG1390
C                                 COEFF CONTAINS THE COEFFICIENTS           DVOG1420
C                                 USED IN SELECTING THE STEP SIZE AND       DVOG1430
C                                 THE ORDER. THESE CONSTANTS NEED ONLY      DVOG1440
C                                 BE ACCURATE TO A FEW DIGITS.              DVOG1450
      DATA                COEF/2.,4.5,7.333,10.42,13.7,17.15,1.,            DVOG1460
     *                    2.0,12.0,24.0,37.89,53.33,70.08,87.97,            DVOG1470
     *                    3.0,6.0,9.167,12.5,15.98,1.0,1.0,                 DVOG1480
     *                    12.0,24.0,37.89,53.33,70.08,87.97,1.0,            DVOG1490
     *                    1.,1.,0.5,0.1667,0.04133,0.008267,1.0,            DVOG1500
     *                    1.0,1.0,2.0,1.0,.3157,.07407,.0139/               DVOG1510
      IER=0                                                                 DVOG1520
      JER=0                                                                 DVOG1530
      N4 = N*N                                                              DVOG1540
      IF(MTH.EQ.0) N4=0                                                     DVOG1550
      N1=N*10+N4                                                            DVOG1560
      N2 = N1 + 1                                                           DVOG1570
      N3 = N                                                                DVOG1580
      N5 = N1 + N                                                           DVOG1590
      N6 = N5 + 1                                                           DVOG1600
      N7=N6+N+N                                                             DVOG1610
      N8=N7+N                                                               DVOG1620
      N9=N8+N                                                               DVOG1630
      N10=N8-1                                                              DVOG1640
      IND1=N4+1                                                             DVOG1650
      IND2=N4+2                                                             DVOG1660
      IND9=N4+9                                                             DVOG1670
      IND10=N4+10                                                           DVOG1680
      IRET = 1                                                              DVOG1690
      KFLAG = 1                                                             DVOG1700
C                                 CHECK MAXDER. IF MTH=0, MAXDER MUST       DVOG1710
C                                 BE LESS THAN 7. MAXDER MUST BE LESS       DVOG1720
C                                 THAN 6, OTHERWISE.                        DVOG1730
      MXDER=MAXDER                                                          DVOG1740
      IF(MTH.EQ.0) GO TO 5                                                  DVOG1750
      IF(MAXDER.LE.6) GO TO 10                                              DVOG1760
      JER=68                                                                DVOG1770
      MXDER=6                                                               DVOG1780
      GO TO 10                                                              DVOG1790
    5 IF(MAXDER.LE.7) GO TO 10                                              DVOG1800
      JER=68                                                                DVOG1810
      MXDER=7                                                               DVOG1820
```

```
   10 IF (JSTART .LE. 0) GO TO 35                                    DVOG1830
C                                   TAKE A STEP CONTINUING FROM THE LAST DVOG1840
C                                   STEP, SAVE INFORMATION FOR A POSSIBLEDVOG1850
C                                   RESTART AND CHECK H FOR A POSSIBLE   DVOG1860
C                                   USER CHANGE. HNEW IS THE PREVIOUS    DVOG1870
C                                   STEP SIZE AND IO IS THE CURRENT ORDERDVOG1880
   15 DO 20  I = 1,N                                                 DVOG1890
         DO 20  J = 1,K                                              DVOG1900
            WK(N4+J,I)=Y(J,I)                                        DVOG1910
   20 CONTINUE                                                       DVOG1920
      HOLD = HNEW                                                    DVOG1930
      IF (H .EQ. HOLD) GO TO 30                                      DVOG1940
   25 RACUM = H/HOLD                                                 DVOG1950
      IRET1 = 1                                                      DVOG1960
      GO TO 375                                                      DVOG1970
   30 IOOLD = IO                                                     DVOG1980
      TOLD = T                                                       DVOG1990
      RACUM = ONE                                                    DVOG2000
      IF (JSTART .GT. 0) GO TO 135                                   DVOG2010
      GO TO 50                                                       DVOG2020
   35 IF (JSTART .EQ. -1) GO TO 45                                   DVOG2030
C                                   FIRST CALL, THE ORDER IS SET TO 1 ANDDVOG2040
C                                   THE INITIAL DERIVATIVES CALCULATED.  DVOG2050
      IO  = 1                                                        DVOG2060
      CALL DFUN(Y,T,N,WK(N2,1),WK,0)                                 DVOG2070
      DO 40  I = 1,N                                                 DVOG2080
         N11 = N1 + I                                                DVOG2090
         Y(2,I)=WK(N11,1)*H                                          DVOG2100
   40 CONTINUE                                                       DVOG2110
      HNEW = H                                                       DVOG2120
      K = 2                                                          DVOG2130
      GO TO 15                                                       DVOG2140
C                                   REPEAT LAST STEP. RESTORE THE SAVED  DVOG2150
C                                   INFORMATION.                         DVOG2160
   45 IF (IO .EQ. IOOLD) JSTART = 1                                  DVOG2170
      T = TOLD                                                       DVOG2180
      IO = IOOLD                                                     DVOG2190
      K = IO + 1                                                     DVOG2200
      GO TO 25                                                       DVOG2210
C                                   SET ALL COEFFICIENTS DETERMINED BY   DVOG2220
C                                   THE ORDER AND THE METHOD TYPE.       DVOG2230
   50 IF (MTH .EQ. 0) GO TO 55                                       DVOG2240
      GO TO (95,100,105,110,115,120),IO                              DVOG2250
   55 GO TO (60,65,70,75,80,85,90),IO                                DVOG2260
   60 C(1) = -ONE                                                    DVOG2270
```

```
      GO TO 125                                                         DVOG2280
   65 C(1)=-HALF                                                        DVOG2290
      C(3)=-HALF                                                        DVOG2300
      GO TO 125                                                         DVOG2310
   70 C(1) = -0.41666666666666667D0                                     DVOG2320
      C(3) = -0.75D0                                                    DVOG2330
      C(4) = -0.16666666666666667D0                                     DVOG2340
      GO TO 125                                                         DVOG2380
   75 C(1) = -0.375D0                                                   DVOG2390
      C(3) = -0.91666666666666667D0                                     DVOG2400
      C(4) = -0.33333333333333333D0                                     DVOG2410
      C(5) = -0.04166666666666667D0                                     DVOG2420
      GO TO 125                                                         DVOG2470
   80 C(1) = -0.34861111111111111D0                                     DVOG2480
      C(3) = -1.04166666666666667D0                                     DVOG2490
      C(4) = -0.48611111111111111D0                                     DVOG2500
      C(5) = -0.10416666666666667D0                                     DVOG2510
      C(6) = -0.00833333333333333333D0                                  DVOG2520
      GO TO 125                                                         DVOG2580
   85 C(1) = -0.32986111111111111D0                                     DVOG2590
      C(3) = -1.14166666666666667D0                                     DVOG2600
      C(4) = -0.625D0                                                   DVOG2610
      C(5) = -0.17708333333333333D0                                     DVOG2620
      C(6) = -0.025D0                                                   DVOG2630
      C(7) = -0.00138888888888888889D0                                  DVOG2640
      GO TO 125                                                         DVOG2710
   90 C(1) = -0.31559193121693120D0                                     DVOG2720
      C(3) = -1.225D0                                                   DVOG2730
      C(4) = -0.75185185185185185190D0                                  DVOG2740
      C(5) = -0.25520833333333333D0                                     DVOG2750
      C(6) = -0.04861111111111111D0                                     DVOG2760
      C(7) = -.48611111111111111D-2                                     DVOG2770
      C(8) = -.19841269841269841D-3                                     DVOG2780
      GO TO 125                                                         DVOG2860
   95 C(1)=-ONE                                                         DVOG2870
      GO TO 125                                                         DVOG2880
  100 C(1) = -0.66666666666666667D0                                     DVOG2890
      C(3) = -0.33333333333333333D0                                     DVOG2900
      GO TO 125                                                         DVOG2930
  105 C(1) = -0.54545454545454545D0                                     DVOG2940
      C(3) = C(1)                                                       DVOG2960
      C(4) = -0.09090909090909091D0                                     DVOG2970
      GO TO 125                                                         DVOG2990
  110 C(1) = -0.48D0                                                    DVOG3000
      C(3) = -0.7D0                                                     DVOG3010
```

```
      C(4) = -0.2D0                                                     DVOG3020
      C(5) = -0.02D0                                                    DVOG3030
      GO TO 125                                                         DVOG3080
  115 C(1) = -0.437956204379562D0                                       DVOG3090
      C(3) = -0.821167883211678 8D0                                     DVOG3100
      C(4) = -0.310218978102189 8D0                                     DVOG3110
      C(5) = -0.0547445255474452 6D0                                    DVOG3120
      C(6) = -0.00364963503649635 04D0                                  DVOG3130
      GO TO 125                                                         DVOG3190
  120 C(1) = -0.408163265306122 5D0                                     DVOG3200
      C(3) = -0.920634920634920 6D0                                     DVOG3210
      C(4) = -0.416666666666666 7D0                                     DVOG3220
      C(5) = -0.0992063492063492D0                                      DVOG3230
      C(6) = -0.0119047619047619D0                                      DVOG3240
      C(7) = -0.00056689342403628 2D0                                   DVOG3250
C                                  IF THE JACOBIAN MUST BE RE-CALCULATEDDVOG3320
C                                  BECAUSE OF AN ORDER CHANGE, SET      DVOG3330
C                                  INEVAL POSITIVE AND REPEAT THE INTE- DVOG3340
C                                  GRATION STEP IF IT HAS NOT YET BEEN  DVOG3350
C                                  DONE(IRET=1) OR SKIP TO A FINAL SCAL-DVOG3360
C                                  ING IF IT HAS BEEN COMPLETED(IRET=2).DVOG3370
C                                  E IS USED FOR COMPARISON OF ERRORS INDVOG3380
C                                  THE CURRENT ORDER. EUP IS USED TO    DVOG3390
C                                  INCREASE THE ORDER, EDWN IS USED TO  DVOG3400
C                                  DECREASE THE ORDER.                  DVOG3410
  125 K = IO+1                                                          DVOG3420
      IDOUB = K                                                         DVOG3430
      MTYP = (4 - MTH)/2                                                DVOG3440
      ENQ2 = HALF/(IO + 1)                                              DVOG3450
      ENQ3 = HALF/(IO + 2)                                              DVOG3460
      ENQ1 = HALF/(IO)                                                  DVOG3470
      PEPSH = EPS                                                       DVOG3480
      EUP = (COEF(IO,MTYP,2)*PEPSH)**2                                  DVOG3490
      E = (COEF(IO,MTYP,1)*PEPSH)**2                                    DVOG3500
      EDWN = (COEF(IO,MTYP,3)*PEPSH)**2                                 DVOG3510
      IF (EDWN .EQ. 0) GO TO 390                                        DVOG3520
      BND = EPS*ENQ3/N                                                  DVOG3530
  130 INEVAL = MTH                                                      DVOG3540
      GO TO (135,340), IRET                                             DVOG3550
C                                  COMPUTE THE PREDICTED Y VALUES BY    DVOG3560
C                                  EFFECTIVELY MULTIPLYING THE SAVED    DVOG3570
C                                  INFORMATION BY THE PASCAL TRIANGLE   DVOG3580
C                                  MATRIX.                              DVOG3590
  135 T = T + H                                                         DVOG3600
      DO 140 J = 2,K                                                    DVOG3610
```

```
      DO 140 J1 = J,K                                                   DVOG3620
         J2 = K - J1 + J - 1                                            DVOG3630
         DO 140 I = 1,N                                                 DVOG3640
            Y(J2,I) = Y(J2,I) + Y(J2+1,I)                               DVOG3650
  140 CONTINUE                                                          DVOG3660
C                                 TAKE UP TO 3 CORRECTOR ITERATIONS.    DVOG3670
C                                 CONVERGENCE IS TESTED BY REQUIRING    DVOG3680
C                                 CHANGES TO BE LESS THAN BND.          DVOG3690
C                                 THE SUMS OF THE CORRECTIONS ARE       DVOG3700
C                                 ACCUMULATED IN THE ARRAY ERROR(I). ITDVOG3710
C                                 IS EQUAL TO THE K-TH DERIVATIVE OF Y DVOG3720
C                                 MULTIPLIED BY H**K/(FACTORIAL(K-1) * DVOG3730
C                                 (C(K)). ERROR(I) IS THEREFORE PRO-    DVOG3740
C                                 PORTIONAL TO THE ACTUAL ERRORS IN THEDVOG3750
C                                 LOWEST POWER OF H PRESENT. (H**K)     DVOG3760
      DO 145 I = 1,N                                                    DVOG3770
         ERROR(I) = ZERO                                                DVOG3780
  145 CONTINUE                                                          DVOG3790
      DO 220 L = 1,3                                                    DVOG3800
         CALL DFUN(Y,T,N,WK(N2,1),WK,0)                                 DVOG3810
C                                 IF THERE HAS BEEN A CHANGE OF ORDER   DVOG3820
C                                 OR THERE HAS BEEN TROUBLE WITH CON-   DVOG3830
C                                 VERGENCE, THE JACOBIAN IS RE-EVALU-   DVOG3840
C                                 ATED PRIOR TO STARTING THE CORRECTOR DVOG3850
C                                 ITERATION IN THE CASE OF STIFF        DVOG3860
C                                 METHODS. INEVAL IS THEN SET TO -1 AS DVOG3870
C                                 AN INDICATOR THAT IT HAS BEEN DONE.  DVOG3880
         IF (INEVAL .LT. 1) GO TO 185                                   DVOG3890
         IF (MTH .EQ. 2) GO TO 165                                      DVOG3900
         CALL DFUN(Y,T,N3,WK,WK,1)                                      DVOG3910
         R = C(1)*H                                                     DVOG3920
         DO 150 I = 1,N4                                                DVOG3930
            WK(I,1)=WK(I,1)*R                                           DVOG3940
  150    CONTINUE                                                       DVOG3950
  155    N11 = N3 + 1                                                   DVOG3960
         N12 = N*N11 - N3                                               DVOG3970
         DO 160 I = 1,N12,N11                                           DVOG3980
            WK(I,1)=WK(I,1)+ONE                                         DVOG3990
  160    CONTINUE                                                       DVOG4000
         INEVAL = -1                                                    DVOG4010
         IF(N.EQ.1) GO TO 185                                           DVOG4020
C                                 LU DECOMPOSE JACOBIAN                 DVOG4030
         CALL LUDAIF(WK,WK,N,N3,NDIG,D1,D2,WK(N7,1),WK(N8,1),WK(N9,1),  DVOG4040
     *      KER)                                                        DVOG4050
         IF (KER) 185,185,225                                           DVOG4060
```

```
165     DO 170 I = 1,N
           WK(IND9,I)=Y(1,I)                                               DVOG4070
170     CONTINUE                                                           DVOG4080
        DO 180 J = 1,N                                                     DVOG4090
           R=EPS*DMAX1(EPS,DABS(WK(IND9,J)))                               DVOG4100
           Y(1,J) = Y(1,J) + R                                             DVOG4110
           D = C(1)*H/R                                                    DVOG4130
           CALL DFUN(Y,T,N,WK(N6,1),WK,0)                                  DVOG4140
           DO 175 I = 1,N                                                  DVOG4150
              N11 = I + (J-1)*N3                                           DVOG4160
              N12 = N5 + I                                                 DVOG4170
              N13 = N1 + I                                                 DVOG4180
              WK(N11,1)=(WK(N12,1)-WK(N13,1))*D                            DVOG4190
175        CONTINUE                                                        DVOG4200
           Y(1,J)=WK(IND9,J)                                               DVOG4210
180     CONTINUE                                                           DVOG4220
        GO TO 155                                                          DVOG4230
185     IF (MTH .NE. 0) GO TO 195                                          DVOG4240
        DO 190 I = 1,N                                                     DVOG4250
           N11 = N1 + I                                                    DVOG4260
           WK(IND9,I)=Y(2,I)-WK(N11,1)*H                                   DVOG4270
190     CONTINUE                                                           DVOG4280
        GO TO 210                                                          DVOG4290
195     DO 200 I = 1,N                                                     DVOG4300
           N11 = N5 + I                                                    DVOG4310
           N12 = N1 + I                                                    DVOG4320
           WK(N11,1)=Y(2,I)-WK(N12,1)*H                                    DVOG4330
200     CONTINUE                                                           DVOG4340
        IF(N.GT.1) GO TO 202                                               DVOG4350
        WK(N8,1) = WK(N6,1) / WK(1,1)                                      DVOG4360
        GO TO 203                                                          DVOG4370
202     CALL LUELMF(WK,WK(N6,1),WK(N7,1),N,N3,WK(N8,1))                    DVOG4380
203     DO 205 I=1,N                                                       DVOG4390
           WK(IND9,I)=WK(N10+I,1)                                          DVOG4400
205     CONTINUE                                                           DVOG4410
210     NT = N                                                             DVOG4420
        DO 215 I = 1,N                                                     DVOG4430
           Y(1,I)=Y(1,I)+C(1)*WK(IND9,I)                                   DVOG4440
           Y(2,I)=Y(2,I)-WK(IND9,I)                                        DVOG4450
           ERROR(I)=ERROR(I)+WK(IND9,I)                                    DVOG4460
           IF (DABS(WK(IND9,I)).LE. (BND*YMAX(I))) NT=NT-1                 DVOG4470
215     CONTINUE                                                           DVOG4480
        IF (NT .LE. 0) GO TO 245                                           DVOG4500
    220 CONTINUE                                                           DVOG4510
C                                    THE CORRECTOR FAILED TO CONVERGE IN 3DVOG4530
```

```
C                                         ITERATIONS. VARIOUS POSSIBILITIES AREDVOG4540
C                                         CHECKED. IF H IS EQUAL TO HMIN AND    DVOG4550
C                                         THIS IS THE ADAMS METHOD OR A STIFF   DVOG4560
C                                         METHOD IN WHICH THE JACOBIAN HAS      DVOG4570
C                                         ALREADY BEEN RE-EVALUATED, A NO CON-  DVOG4580
C                                         VERGENCE EXIT IS TAKEN. OTHERWISE THEDVOG4590
C                                         JACOBIAN IS RE-EVALUATED AND/OR THE   DVOG4600
C                                         STEP IS REDUCED TO TRY TO OBTAIN      DVOG4610
C                                         CONVERGENCE.                          DVOG4620
  225 T = T - H                                                                 DVOG4630
      IF ((DABS(H) .LE. (DABS(HMIN)*ONEP)) .AND. ((IWEVAL - MTYP) .LT.         DVOG4640
     *    -1)) GO TO 230                                                        DVOG4650
      IF ((MTH .EQ. 0) .OR. (IWEVAL .NE. 0)) RACUM = RACUM*0.25                 DVOG4680
      IWEVAL = MTH                                                              DVOG4690
      IRET1 = 2                                                                 DVOG4700
      GO TO 375                                                                 DVOG4710
  230 KFLAG = -3                                                                DVOG4720
  235 DO 240 I = 1,N                                                            DVOG4730
         DO 240 J = 1,K                                                         DVOG4740
            Y(J,I)=WK(N4+J,I)                                                   DVOG4750
  240 CONTINUE                                                                  DVOG4760
      H = HOLD                                                                  DVOG4770
      IO = IOOLD                                                                DVOG4780
      JSTART = IO                                                               DVOG4790
      GO TO 395                                                                 DVOG4800
C                                         THE CORRECTOR CONVERGED AND CONTROL   DVOG4810
C                                         IS PASSED TO STATEMENT 260 IF THE     DVOG4820
C                                         ERROR TEST IS PASSED, AND TO 270      DVOG4830
C                                         OTHERWISE.                            DVOG4840
C                                         IF THE STEP IS O.K. IT IS ACCEPTED.   DVOG4850
C                                         IF IDOUB HAS BEEN REDUCED TO ONE,     DVOG4860
C                                         A TEST IS MADE TO SEE IF THE STEP CANDVOG4870
C                                         BE INCREASED AT ONE LOWER, THE        DVOG4880
C                                         CURRENT, OR ONE HIGHER ORDER. A STEP  DVOG4890
C                                         CHANGE IS ONLY MADE IF THE STEP CAN   DVOG4900
C                                         BE INCREASED BY AT LEAST 1.1*H. IF NODVOG4910
C                                         CHANGE IS POSSIBLE, IDOUB IS SET TO   DVOG4920
C                                         10 TO PREVENT FURTHER TESTING FOR THEDVOG4930
C                                         NEXT TEN STEPS.                       DVOG4940
C                                         IF A CHANGE IS POSSIBLE, IT IS MADE   DVOG4950
C                                         AND IDOUB IS SET TO IO + 1 TO PREVENTDVOG4960
C                                         FURTHER TESTING FOR THAT NUMBER OF    DVOG4970
C                                         STEPS. IF THE ERROR WAS TOO LARGE,    DVOG4980
C                                         THE OPTIMUM STEP SIZE FOR THIS OR     DVOG4990
C                                         SOME LOWER ORDER IS COMPUTED, AND THEDVOG5000
```

```
C                                          STEP RETRIED. IF IT SHOULD FAIL TWICEDVOG5010
C                                          MORE IT IS AN INDICATION THAT THE     DVOG5020
C                                          DERIVATIVES THAT HAVE ACCUMULATED IN DVOG5030
C                                          THE Y ARRAY HAVE ERRORS OF THE WRONG DVOG5040
C                                          ORDER, SO THE FIRST DERIVATIVES ARE  DVOG5050
C                                          RECOMPUTED AND THE ORDER IS SET TO 1.DVOG5060
  245 D = ZERO                                                             DVOG5070
      DO 250 I = 1,N                                                       DVOG5080
        D = D + (ERROR(I)/YMAX(I))**2                                      DVOG5090
  250 CONTINUE                                                             DVOG5100
      IWEVAL = 0                                                           DVOG5110
      IF (D .GT. E) GO TO 270                                              DVOG5120
      IF (K .LT. 3) GO TO 260                                              DVOG5130
      DO 255 J = 3,K                                                       DVOG5140
        DO 255 I = 1,N                                                     DVOG5150
              Y(J,I) = Y(J,I) + C(J)*ERROR(I)                              DVOG5160
  255 CONTINUE                                                             DVOG5170
  260 KFLAG = +1                                                           DVOG5180
      HNEW = H                                                             DVOG5190
      IF (IDOUB .LE. 1) GO TO 275                                          DVOG5200
      IDOUB = IDOUB - 1                                                    DVOG5210
      IF (IDOUB .GT. 1) GO TO 350                                          DVOG5220
      DO 265 I = 1,N                                                       DVOG5230
        WK(IND10,I)=ERROR(I)                                               DVOG5240
  265 CONTINUE                                                             DVOG5250
      GO TO 350                                                            DVOG5260
  270 KFLAG = KFLAG - 2                                                    DVOG5270
      IF (DABS(H) .LE. (DABS(HMIN)*ONEP)) GO TO 370                        DVOG5280
      T = TOLD                                                             DVOG5300
      IF (KFLAG .LE. -5) GO TO 360                                         DVOG5310
  275 PR2 = (D/E)**ENQ2*1.2                                                DVOG5320
      PR3 = 1.E+20                                                         DVOG5330
      IF ((IQ .GE. MXDER) .OR. (KFLAG .LE. -1)) GO TO 285                  DVOG5340
      D = ZERO                                                             DVOG5350
      DO 280 I = 1,N                                                       DVOG5360
        D=D+((ERROR(I)-WK(IND10,I))/YMAX(I))**2                            DVOG5370
  280 CONTINUE                                                             DVOG5380
      PR3 = (D/EUP)**ENQ3*1.4                                              DVOG5390
  285 PR1 = 1.E+20                                                         DVOG5400
      IF (IQ .LE. 1) GO TO 295                                             DVOG5410
      D = ZERO                                                             DVOG5420
      DO 290 I = 1,N                                                       DVOG5430
        D = D + (Y(K,I)/YMAX(I))**2                                        DVOG5440
  290 CONTINUE                                                             DVOG5450
      PR1 = (D/EDWN)**ENQ1*1.3                                             DVOG5460
```

```
  295 CONTINUE                                                          DVOG5470
      IF (PR2 .LE. PR3) GO TO 325                                       DVOG5480
      IF (PR3 .LT. PR1) GO TO 330                                       DVOG5490
  300 R = 1.0/AMAX1(PR1,1.E-4)                                          DVOG5500
      NEWI = IO - 1                                                     DVOG5510
  305 IDOUB = 10                                                        DVOG5520
      IF ((KFLAG .EQ. 1) .AND. (R .LT. (1.1))) GO TO 350                DVOG5530
      IF (NEWI .LE. IO) GO TO 315                                       DVOG5540
      XK = ONE / K                                                      DVOG5550
      DO 310 I = 1,N                                                    DVOG5560
         Y(NEWI+1,I) = ERROR(I)*C(K)*XK                                 DVOG5570
  310 CONTINUE                                                          DVOG5580
  315 K = NEWI + 1                                                      DVOG5590
      IF (KFLAG .EQ. 1) GO TO 335                                       DVOG5600
      RACUM = RACUM*R                                                   DVOG5610
      IRET1 = 3                                                         DVOG5620
      GO TO 375                                                         DVOG5630
  320 IF (NEWI .EQ. IO) GO TO 135                                       DVOG5640
      IO = NEWI                                                         DVOG5650
      GO TO 50                                                          DVOG5660
  325 IF (PR2 .GT. PR1) GO TO 300                                       DVOG5670
      NEWI = IO                                                         DVOG5680
      R = 1.0/AMAX1(PR2,1.E-4)                                          DVOG5690
      GO TO 305                                                         DVOG5700
  330 R = 1.0/AMAX1(PR3,1.E-4)                                          DVOG5710
      NEWI = IO + 1                                                     DVOG5720
      GO TO 305                                                         DVOG5730
  335 IRET = 2                                                          DVOG5740
      R = DMIN1(R,DABS(HMAX/H))                                         DVOG5750
      H = H*R                                                           DVOG5770
      HNEW = H                                                          DVOG5780
      IF (IO .EQ. NEWI) GO TO 340                                       DVOG5790
      IO = NEWI                                                         DVOG5800
      GO TO 50                                                          DVOG5810
  340 R1 = ONE                                                          DVOG5820
      DO 345 J = 2,K                                                    DVOG5830
         R1 = R1*R                                                      DVOG5840
         DO 345 I = 1,N                                                 DVOG5850
               Y(J,I) = Y(J,1)*R1                                       DVOG5860
  345 CONTINUE                                                          DVOG5870
      IDOUB = K                                                         DVOG5880
  350 DO 355 I = 1,N                                                    DVOG5890
         YMAX(I) = DMAX1(YMAX(I),DABS(Y(1,I)))                          DVOG5900
  355 CONTINUE                                                          DVOG5920
      JSTART = IO                                                       DVOG5930
```

```
      GO TO 395                                                         DVOG5940
  360 IF (IO .EQ. 1) GO TO 390                                          DVOG5950
      CALL DFUN(Y,T,N,WK(N2,1),WK,0)                                    DVOG5960
      R = H/HOLD                                                        DVOG5970
      DO 365 I = 1,N                                                    DVOG5980
         Y(1,I)=WK(IND1,I)                                              DVOG5990
         N11 = N1 + I                                                   DVOG6000
         WK(IND2,I)=HOLD*WK(N11,1)                                      DVOG6010
         Y(2,I)=WK(IND2,I)*R                                            DVOG6020
  365 CONTINUE                                                          DVOG6030
      IO = 1                                                            DVOG6040
      KFLAG = 1                                                         DVOG6050
      GO TO 50                                                          DVOG6060
  370 KFLAG = -1                                                        DVOG6070
      HNEW = H                                                          DVOG6080
      JSTART = IO                                                       DVOG6090
      GO TO 400                                                         DVOG6100
C                                  SCALE ALL VARIABLES CONNECTED WITH   DVOG6110
C                                  H AND RETURN TO THE CALLING SECTION  DVOG6120
  375 RACUM = DMAX1(DABS(HMIN/HOLD),RACUM)                              DVOG6130
      RACUM = DMIN1(RACUM,DABS(HMAX/HOLD))                              DVOG6140
      R1 = ONE                                                          DVOG6170
      DO 380 J = 2,K                                                    DVOG6180
         R1 = R1*RACUM                                                  DVOG6190
         DO 380 I = 1,N                                                 DVOG6200
            Y(J,I)=WK(N4+J,I)*R1                                        DVOG6210
  380 CONTINUE                                                          DVOG6220
      H = HOLD*RACUM                                                    DVOG6230
      DO 385 I = 1,N                                                    DVOG6240
         Y(1,I)=WK(IND1,I)                                              DVOG6250
  385 CONTINUE                                                          DVOG6260
      IDOUB = K                                                         DVOG6270
      GO TO (30,135,320),IRET1                                          DVOG6280
  390 KFLAG = -4                                                        DVOG6290
      GO TO 235                                                         DVOG6300
  395 IF(KFLAG.EQ.1) GO TO 9000                                         DVOG6310
  400 IER=32-KFLAG                                                      DVOG6320
 9000 CONTINUE                                                          DVOG6330
      IF(JER.NE.0) CALL UERTST(JER,6HDVOGER)                            DVOG6340
      IF(IER.GT.33) IER = IER - 1                                       DVOG6350
      IF(IER.NE.0) CALL UERTST(IER,6HDVOGER)                            DVOG6360
 9005 RETURN                                                            DVOG6370
      END                                                               DVOG6380
```

```
      SUBROUTINE LUDATF (A,LU,N,IA,IDGT,D1,D2,IPVT,EQUIL,WA,IER)         LUDA0670
C                                                                       LUDA0020
C-LUDATF--------D--------LIBRARY 1--------------------------------------LUDA0030
C                                                                       LUDA0040
C   FUNCTION            - L-U DECOMPOSITION BY THE CROUT ALGORITHM      LUDA0050
C                           WITH OPTIONAL ACCURACY TEST.                LUDA0060
C   USAGE               - CALL LUDATF(A,LU,N,IA,IDGT,D1,D2,IPVT,        LUDA0070
C                                EQUIL,WA,IER)                          LUDA0080
C   PARAMETERS   A      - INPUT MATRIX OF DIMENSION N BY N CONTAINING   LUDA0090
C                           THE MATRIX TO BE DECOMPOSED                 LUDA0100
C                LU     - REAL OUTPUT MATRIX OF DIMENSION N BY N        LUDA0110
C                           CONTAINING THE L-U DECOMPOSITION OF A       LUDA0120
C                           ROWWISE PERMUTATION OF THE INPUT MATRIX.    LUDA0130
C                           FOR A DESCRIPTION OF THE FORMAT OF LU, SEE  LUDA0140
C                           EXAMPLE.                                    LUDA0150
C                N      - INPUT SCALAR CONTAINING THE ORDER OF THE      LUDA0160
C                           MATRIX A.                                   LUDA0170
C                IA     - INPUT SCALAR CONTAINING THE ROW DIMENSION OF  LUDA0180
C                           MATRICES A AND LU IN THE CALLING PROGRAM.   LUDA0190
C                IDGT   - INPUT OPTION.                                 LUDA0200
C                           IF IDGT IS GREATER THAN ZERO, THE NON-ZERO  LUDA0210
C                           ELEMENTS OF A ARE ASSUMED TO BE CORRECT TO  LUDA0220
C                           IDGT DECIMAL PLACES.  LUDATF PERFORMS AN    LUDA0230
C                           ACCURACY TEST TO DETERMINE IF THE COMPUTED  LUDA0240
C                           DECOMPOSITION IS THE EXACT DECOMPOSITION    LUDA0250
C                           OF A MATRIX WHICH DIFFERS FROM THE GIVEN ONELUDA0260
C                           BY LESS THAN ITS UNCERTAINTY.               LUDA0270
C                         IF IDGT IS EQUAL TO ZERO, THE ACCURACY TEST ISLUDA0280
C                           BYPASSED.                                   LUDA0290
C                D1     - OUTPUT SCALAR CONTAINING ONE OF THE TWO       LUDA0300
C                           COMPONENTS OF THE DETERMINANT. SEE          LUDA0310
C                           DESCRIPTION OF PARAMETER D2, BELOW.         LUDA0320
C                D2     - OUTPUT SCALAR CONTAINING ONE OF THE           LUDA0330
C                           TWO COMPONENTS OF THE DETERMINANT. THE      LUDA0340
C                           DETERMINANT MAY BE EVALUATED AS (D1)(2**D2) LUDA0350
C                IPVT   - OUTPUT VECTOR OF LENGTH N CONTAINING THE      LUDA0360
C                           PERMUTATION INDICES. SEE DOCUMENT           LUDA0370
C                           (ALGORITHM).                                LUDA0380
C                EQUIL  - OUTPUT VECTOR OF LENGTH N CONTAINING          LUDA0390
C                           RECIPROCALS OF THE ABSOLUTE VALUES OF       LUDA0400
C                           THE LARGEST (IN ABSOLUTE VALUE) ELEMENT     LUDA0410
C                           IN EACH ROW.                                LUDA0420
C                WA     - ACCURACY TEST PARAMETER, OUTPUT ONLY IF       LUDA0430
C                           IDGT IS GREATER THAN ZERO.                  LUDA0440
C                           SEE ELEMENT DOCUMENTATION FOR DETAILS.      LUDA0450
```

```
C                  IER    - ERROR PARAMETER                             LUDA0460
C                           TERMINAL ERROR=128+N                        LUDA0470
C                             N = 1 INDICATES THAT MATRIX A IS          LUDA0480
C                                   ALGORITHMICALLY SINGULAR. (SEE THE  LUDA0490
C                                   CHAPTER L PRELUDE).                 LUDA0500
C                           WARNING ERROR=32+N                          LUDA0510
C                             N = 2 INDICATES THAT THE ACCURACY TEST    LUDA0520
C                                   FAILED.                             LUDA0530
C                                   THE COMPUTED SOLUTION MAY BE IN ERROR LUDA0540
C                                   BY MORE THAN CAN BE ACCOUNTED FOR BY LUDA0550
C                                   THE UNCERTAINTY OF THE DATA.        LUDA0560
C                                   THIS WARNING CAN BE PRODUCED ONLY IF LUDA0570
C                                   IDGT IS GREATER THAN 0 ON INPUT.    LUDA0580
C                                   SEE CHAPTER L PRELUDE FOR FURTHER   LUDA0590
C                                   DISCUSSION.                         LUDA0600
C   PRECISION             - SINGLE/DOUBLE                               LUDA0610
C   REQD. IMSL ROUTINES   - UERTST                                      LUDA0620
C   LANGUAGE              - FORTRAN                                     LUDA0630
C-----------------------------------------------------------------------LUDA0640
C   LATEST REVISION       - AUGUST 15, 1973                             LUDA0650
C                                                                       LUDA0660
C                                                                       LUDA0680
      DIMENSION          A(IA,N),LU(IA,N),IPVT(N),EQUIL(N)              LUDA0690
      DOUBLE PRECISION   A,LU,D1,D2,EQUIL,WA,ZERO,ONE,FOUR,SIXTN,SIXTH, LUDA0700
     *                   RN,WREL,BIGA,BIG,P,SUM,AI,WI,T,TEST,Q          LUDA0710
     2                  ,DABS
      DATA               ZERO,ONE,FOUR,SIXTN,SIXTH/0.D0,1.D0,4.D0,      LUDA0730
     *                   16.D0,.0625D0/                                 LUDA0740
C                                  INITIALIZATION                       LUDA0760
      IER = 0                                                           LUDA0770
      RN = N                                                            LUDA0780
      WREL = ZERO                                                       LUDA0790
      D1 = ONE                                                          LUDA0800
      D2 = ZERO                                                         LUDA0810
      BIGA = ZERO                                                       LUDA0820
      DO 10 I=1,N                                                       LUDA0830
         BIG = ZERO                                                     LUDA0840
         DO 5 J=1,N                                                     LUDA0850
            P = A(I,J)                                                  LUDA0860
            LU(I,J) = P                                                 LUDA0870
            P = DABS(P)                                                 LUDA0880
            IF (P .GT. BIG) BIG = P                                     LUDA0900
    5    CONTINUE                                                       LUDA0910
         IF (BIG .GT. BIGA) BIGA = BIG                                  LUDA0920
         IF (BIG .EQ. ZERO) GO TO 110                                   LUDA0930
```

```
      EQUIL(I) = ONE/BIG                                                LUDA0940
   10 CONTINUE                                                          LUDA0950
      DO 105 J=1,N                                                      LUDA0960
         JM1 = J-1                                                      LUDA0970
         IF (JM1 .LT. 1) GO TO 40                                       LUDA0980
C                                  COMPUTE U(I,J), I=1,...,J-1          LUDA0990
         DO 35 I=1,JM1                                                  LUDA1000
            SUM = LU(I,J)                                               LUDA1010
            IM1 = I-1                                                   LUDA1020
            IF (IDGT .EQ. 0) GO TO 25                                   LUDA1030
C                                  WITH ACCURACY TEST                   LUDA1040
            AI = DABS(SUM)                                              LUDA1050
            WI = ZERO                                                   LUDA1070
            IF (IM1 .LT. 1) GO TO 20                                    LUDA1080
            DO 15 K=1,IM1                                               LUDA1090
               T = LU(I,K)*LU(K,J)                                      LUDA1100
               SUM = SUM-T                                              LUDA1110
               WI = WI+DABS(T)                                          LUDA1120
   15       CONTINUE                                                    LUDA1140
            LU(I,J) = SUM                                               LUDA1150
   20       WI = WI+DABS(SUM)                                           LUDA1160
            IF (AI .EQ. ZERO) AI = BIGA                                 LUDA1180
            TEST = WI/AI                                                LUDA1190
            IF (TEST .GT. WREL) WREL = TEST                             LUDA1200
            GO TO 35                                                    LUDA1210
C                                  WITHOUT ACCURACY                     LUDA1220
   25       IF (IM1 .LT. 1) GO TO 35                                    LUDA1230
            DO 30 K=1,IM1                                               LUDA1240
               SUM = SUM-LU(I,K)*LU(K,J)                                LUDA1250
   30       CONTINUE                                                    LUDA1260
            LU(I,J) = SUM                                               LUDA1270
   35    CONTINUE                                                       LUDA1280
   40    P = ZERO                                                       LUDA1290
C                                  COMPUTE U(J,J) AND L(I,J), I=J+1,...,LUDA1300
         DO 70 I=J,N                                                    LUDA1310
            SUM = LU(I,J)                                               LUDA1320
            IF (IDGT .EQ. 0) GO TO 55                                   LUDA1330
C                                  WITH ACCURACY TEST                   LUDA1340
            AI = DABS(SUM)                                              LUDA1350
            WI = ZERO                                                   LUDA1370
            IF (JM1 .LT. 1) GO TO 50                                    LUDA1380
            DO 45 K=1,JM1                                               LUDA1390
               T = LU(I,K)*LU(K,J)                                      LUDA1400
               SUM = SUM-T                                              LUDA1410
               WI = WI+DABS(T)                                          LUDA1420
```

```
C     45      CONTINUE                                                   LUDA1440
              LU(I,J) = SUM                                              LUDA1450
      50      WI = WI+DABS(SUM)                                          LUDA1460
              IF (AI .EQ. ZERO) AI = BIGA                                LUDA1480
              TEST = WI/AI                                               LUDA1490
              IF (TEST .GT. WREL) WREL = TEST                            LUDA1500
              GO TO 65                                                   LUDA1510
C                                  WITHOUT ACCURACY TEST                 LUDA1520
      55      IF (JM1 .LT. 1) GO TO 65                                   LUDA1530
              DO 60 K=1,JM1                                              LUDA1540
                 SUM = SUM-LU(I,K)*LU(K,J)                               LUDA1550
      60      CONTINUE                                                   LUDA1560
              LU(I,J) = SUM                                              LUDA1570
      65      Q = EQUIL(I)*DABS(SUM)                                     LUDA1580
              IF (P .GE. Q) GO TO 70                                     LUDA1600
              P = Q                                                      LUDA1610
              IMAX = I                                                   LUDA1620
      70   CONTINUE                                                      LUDA1630
C                                  TEST FOR ALGORITHMIC SINGULARITY      LUDA1640
           IF (RN+P .EQ. RN) GO TO 110                                   LUDA1650
           IF (J .EQ. IMAX) GO TO 80                                     LUDA1660
C                                  INTERCHANGE ROWS J AND IMAX           LUDA1670
           D1 = -D1                                                      LUDA1680
           DO 75 K=1,N                                                   LUDA1690
              P = LU(IMAX,K)                                             LUDA1700
              LU(IMAX,K) = LU(J,K)                                       LUDA1710
              LU(J,K) = P                                                LUDA1720
      75   CONTINUE                                                      LUDA1730
           EQUIL(IMAX) = EQUIL(J)                                        LUDA1740
      80   IPVT(J) = IMAX                                                LUDA1750
           D1 = D1*LU(J,J)                                               LUDA1760
      85   IF (DABS(D1) .LE. ONE) GO TO 90                               LUDA1770
           D1 = D1*SIXTH                                                 LUDA1790
           D2 = D2+FOUR                                                  LUDA1800
           GO TO 85                                                      LUDA1810
      90   IF (DABS(D1) .GE. SIXTH) GO TO 95                             LUDA1820
           D1 = D1*SIXTN                                                 LUDA1840
           D2 = D2-FOUR                                                  LUDA1850
           GO TO 90                                                      LUDA1860
      95   CONTINUE                                                      LUDA1870
           JP1 = J+1                                                     LUDA1880
           IF (JP1 .GT. N) GO TO 105                                     LUDA1890
C                                  DIVIDE BY PIVOT ELEMENT U(J,J)        LUDA1900
           P = LU(J,J)                                                   LUDA1910
           DO 100 I=JP1,N                                                LUDA1920
```

AD-A064 545 CINCINNATI UNIV OH DEPT OF ENGINEERING SCIENCE F/G 12/1
A CRITICAL EVALUATION OF COMPUTER SUBROUTINES FOR SOLVING STIFF--ETC(U)
OCT 78 D C KRINKE, R L HUSTON N00014-76-C-0139
UNCLASSIFIED UC-ES-101578-8-ONR NL

2 OF 2
AD
A064545

END
DATE
FILMED
4-79
DDC

```
                 LU(I,J) = LU(I,J)/P                                          LUDA1930
    100       CONTINUE                                                        LUDA1940
    105 CONTINUE                                                              LUDA1950
C                                      PERFORM ACCURACY TEST                  LUDA1960
        IF (IDGT .EQ. 0) GO TO 9005                                           LUDA1970
        P = 3*N+3                                                             LUDA1980
        WA = P*WREL                                                           LUDA1990
        IF (WA+10.D0**(-IDGT) .NE. WA) GO TO 9005                             LUDA2000
        IER = 34                                                              LUDA2020
        GO TO 9000                                                            LUDA2030
C                                      ALGORITHMIC SINGULARITY                LUDA2040
    110 IER = 129                                                             LUDA2050
        D1 = ZERO                                                             LUDA2060
        D2 = ZERO                                                             LUDA2070
   9000 CONTINUE                                                              LUDA2080
C                                      PRINT ERROR                            LUDA2090
        CALL UERTST(IER,6HLUDATF)                                             LUDA2100
   9005 RETURN                                                                LUDA2110
        END                                                                   LUDA2120
```

```
      SUBROUTINE LUELMF (A,B,IPVT,N,IA,X)                                 LUEF0280
C                                                                         LUEF0020
C-LUELMF--------D-------LIBRARY 1----------------------------------------LUEF0030
C                                                                         LUEF0040
C   FUNCTION            - ELIMINATION PART OF SOLUTION OF AX=B -          LUEF0050
C                           FULL STORAGE MODE                             LUEF0060
C   USAGE               - CALL LUELMF (A,B,IPVT,N,IA,X)                   LUEF0070
C   PARAMETERS   A      - THE RESULT, LU, COMPUTED IN THE SUBROUTINE      LUEF0080
C                           'LUDATF', WHERE L IS A LOWER TRIANGULAR       LUEF0090
C                           MATRIX WITH ONES ON THE MAIN DIAGONAL. U IS   LUEF0100
C                           UPPER TRIANGULAR. L AND U ARE STORED AS A     LUEF0110
C                           SINGLE MATRIX A, AND THE UNIT DIAGONAL OF     LUEF0120
C                           L IS NOT STORED                               LUEF0130
C                B      - B IS A VECTOR OF LENGTH N ON THE RIGHT HAND     LUEF0140
C                           SIDE OF THE EQUATION AX=B                     LUEF0150
C                IPVT   - THE PERMUTATION MATRIX RETURNED FROM THE        LUEF0160
C                           SUBROUTINE 'LUDATF', STORED AS AN N LENGTH    LUEF0170
C                           VECTOR                                        LUEF0180
C                N      - ORDER OF A AND NUMBER OF ROWS IN B              LUEF0190
C                IA     - NUMBER OF ROWS IN THE DIMENSION STATEMENT       LUEF0200
C                           FOR A IN THE CALLING PROGRAM.                 LUEF0210
C                X      - THE RESULT X                                    LUEF0220
C   PRECISION           - SINGLE/DOUBLE                                   LUEF0230
C   LANGUAGE            - FORTRAN                                         LUEF0240
C-----------------------------------------------------------------------LUEF0250
C   LATEST REVISION     - APRIL 11,1975                                   LUEF0260
C                                                                         LUEF0270
C                                                                         LUEF0290
      DIMENSION          A(IA,N),B(N),IPVT(N),X(N)                        LUEF0300
      DOUBLE PRECISION   A,B,X,SUM                                        LUEF0310
C                                  SOLVE LY = B FOR Y                     LUEF0320
      DO 5 I=1,N                                                          LUEF0330
    5 X(I) = B(I)                                                         LUEF0340
      IW = 0                                                              LUEF0350
      DO 20 I=1,N                                                         LUEF0360
         IP = IPVT(I)                                                     LUEF0370
         SUM = X(IP)                                                      LUEF0380
         X(IP) = X(I)                                                     LUEF0390
         IF (IW .EQ. 0) GO TO 15                                          LUEF0400
         IM1 = I-1                                                        LUEF0410
         DO 10 J=IW,IM1                                                   LUEF0420
            SUM = SUM-A(I,J)*X(J)                                         LUEF0430
   10    CONTINUE                                                         LUEF0440
         GO TO 20                                                         LUEF0450
   15    IF (SUM .NE. 0.) IW = I                                          LUEF0460
```

```
   20 X(I) = SUM                                              LUEF0470
C                                     SOLVE JX = Y FOR X      LUEF0480
      DO 30 IB=1,N                                            LUEF0490
         I = N+1-IB                                           LUEF0500
         IP1 = I+1                                            LUEF0510
         SUM = X(I)                                           LUEF0520
         IF (IP1 .GT. N) GO TO 30                             LUEF0530
         DO 25 J=IP1,N                                        LUEF0540
            SUM = SUM-A(I,J)*X(J)                             LUEF0550
   25    CONTINUE                                             LUEF0560
   30 X(I) = SUM/A(I,I)                                       LUEF0570
      RETURN                                                  LUEF0580
      END                                                     LUEF0590
```

```
      SUBROUTINE UERTST(IER,NAME)                                       UERT0190
C                                                                       UERT0020
C-UERTST---------------LIBRARY 1----------------------------------------UERT0030
C                                                                       UERT0040
C   FUNCTION            - ERROR MESSAGE GENERATION                      UERT0050
C   USAGE               - CALL UERTST(IER,NAME)                         UERT0060
C   PARAMETERS   IER    - ERROR PARAMETER. TYPE + N  WHERE              UERT0070
C                            TYPE= 128 IMPLIES TERMINAL ERROR           UERT0080
C                                   64 IMPLIES WARNING WITH FIX         UERT0090
C                                   32 IMPLIES WARNING                  UERT0100
C                            N    = ERROR CODE RELEVANT TO CALLING ROUTINEUERT0110
C                NAME   - INPUT VECTOR CONTAINING THE NAME OF THE       UERT0120
C                            CALLING ROUTINE AS A SIX CHARACTER LITERAL UERT0130
C                            STRING.                                    UERT0140
C   LANGUAGE            - FORTRAN                                       UERT0150
C-----------------------------------------------------------------------UERT0160
C   LATEST REVISION     - JANUARY 18, 1974                              UERT0170
C                                                                       UERT0180
C                                                                       UERT0200
      DIMENSION          ITYP(5,4),IBIT(4)                              UERT0210
      INTEGER*2          NAME(3)                                        UERT0220
      INTEGER            WARN,WARF,TERM,PRINTR                          UERT0230
      EQUIVALENCE        (IBIT(1),WARN),(IBIT(2),WARF),(IBIT(3),TERM)   UERT0240
      DATA      ITYP     /'WARN','ING ','    ','    ','    ',           UERT0250
     *                    'WARN','ING(','WITH',' FIX',')   ',           UERT0260
     *                    'TERM','INAL','    ','    ','    ',           UERT0270
     *                    'NON-','DEFI','NED ','    ','    '/,          UERT0280
     *          IBIT     / 32,64,128,0/                                 UERT0290
      DATA      PRINTR   / 6/                                           UERT0300
      IER2=IER                                                          UERT0310
      IF (IER2 .GE. WARN) GO TO 5                                       UERT0320
C                                    NON-DEFINED                        UERT0330
      IER1=4                                                            UERT0340
      GO TO 20                                                          UERT0350
    5 IF (IER2 .LT. TERM) GO TO 10                                      UERT0360
C                                    TERMINAL                           UERT0370
      IER1=3                                                            UERT0380
      GO TO 20                                                          UERT0390
   10 IF (IER2 .LT. WARF) GO TO 15                                      UERT0400
C                                    WARNING(WITH FIX)                  UERT0410
      IER1=2                                                            UERT0420
      GO TO 20                                                          UERT0430
C                                    WARNING                            UERT0440
   15 IER1=1                                                            UERT0450
C                                    EXTRACT 'N'                        UERT0460
```

```
   20 IER2=IER2-IBIT(IER1)                                               UERT0470
C                                    PRINT ERROR MESSAGE                 UERT0480
      WRITE (PRINTR,25) (ITYP(I,IER1),I=1,5),NAME,IER2,IER               UERT0490
   25 FORMAT(' *** I M S L(UERTST) ***  ',5A4,4X,3A2,4X,I2,              UERT0500
     *   ' (IER = ',I3,')')                                              UERT0510
      RETURN                                                             UERT0520
      END                                                                UERT0530
```

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

(9) Technical rept. 1 Sep 77-15 Oct 78

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER (14) ~~ONR~~-UC-ES-101578-8-ONR	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) (6) A Critical Evaluation of Computer Subroutines for Solving Stiff Differential Equations		5. TYPE OF REPORT & PERIOD COVERED Technical 9/1/77-10/15/78
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) (10) Dennis C./Krinke Ronald L./Huston		8. CONTRACT OR GRANT NUMBER(s) (15) N00014-76-C-0139
9. PERFORMING ORGANIZATION NAME AND ADDRESS University of Cincinnati Cincinnati, Ohio 45221		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS ~~122303~~
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research Resident Representative Purdue University, Room 84 Graduate House Lafayette, Indiana 47907		12. REPORT DATE ~~10/15/78~~ (11) 15 Oct 78
		13. NUMBER OF PAGES (12) 102p.
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Office of Naval Research Structural Mechanics Department of the Navy Arlington, Virginia 22217		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Distribution of this report is unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Differential Equation Solvers, Computer Integrator Subroutines, Systems of Differential Equations, Computer Modelling, Stiff Systems

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A number of commonly available computer subroutines for solving differential equation systems are tested and compared for their ability to solve "stiff" systems.

A "stiff" system is defined as either: 1) a system with widely separated eigenvalues or time constants or 2) a system with diverging exponential terms which have small or zero coefficients due to a particular

next page

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

S/N 0102-LF-014-6601

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

410649

choice of initial conditions. For example, it is believed that the governing differential equations of large complex mechanical system models (such as, human-body/crash-victim simulation models, finite segment structural models, and large vibrating system models) are frequently stiff.

The solver subroutines tested on such systems are as follows: 1) DRKGS (a fourth-order Runge-Kutta routine); 2) DHPCG (a Hamming predictor-corrector routine; 3) DVOGER-ADAMS (an Adams predictor-corrector routine); 4) DVOGER-GEAR (a Gear predictor-corrector routine) 5) DREBS (a Bulirsch-Stoer routine); and 6) RK45 (a sixth-order Runge-Kutta routine).

These solver subroutines are tested and compared for a number of stiff systems of both types described above where the exact analytical solution is known. The subroutines are compared in terms of run time, accuracy, and function calls. It is found that the subroutines vary considerably in terms of accuracy. Also, their overall individual effectiveness is highly dependent upon the specific system being solved. However, for systems of the first type of stiffness, Gear's method (DVOGER-GEAR) and DRKGS appear to be the preferred routines. Finally, the automatic stepsize capability of DRKGS and DHPCG is a definite advantage over the other routines. However, the step oriented format of DVOGER and DREBS provides a potential for similar flexibility in error control through coding modifications.